

# 第4章 组合逻辑电路

## 第一讲：组合逻辑电路的分析和设计

# 教学内容

§ 4.1 概述

§ 4.2 组合逻辑电路的分析方法

§ 4.3 组合逻辑电路的基本设计方法

§ 4.4 常用组合逻辑电路模块

§ 4.5 层次化和模块化设计方法

§ 4.6 可编程逻辑器件

§ 4.9 组合逻辑电路的竞争—冒险现象

## 本章重点

1. 组合逻辑电路的分析与设计方法
2. 常用组合逻辑模块的使用

## 4.1 概述

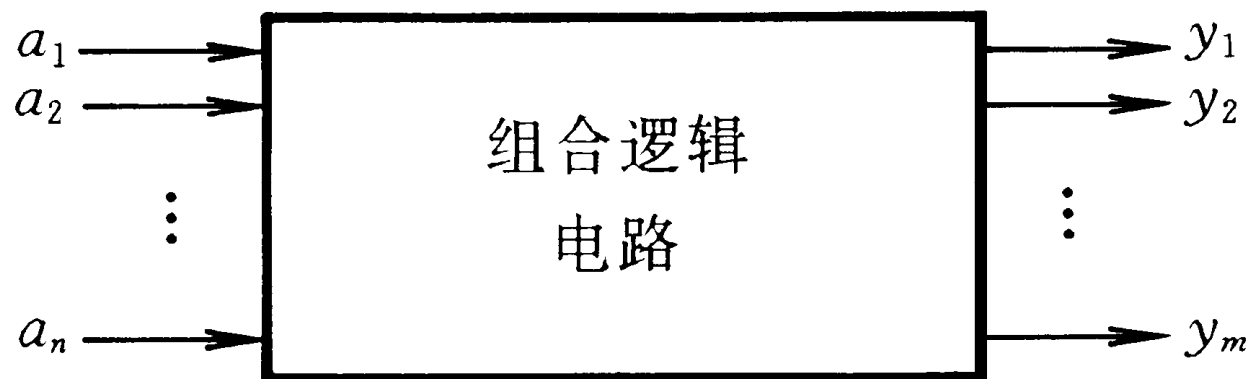
### 数字电路

#### 组合逻辑电路

任一时刻的输出仅取决于该时刻的输入，与电路原来的状态无关。

#### 时序逻辑电路

任一时刻的输出不仅取决于现时的输入，而且还与电路原来状态有关。



组合逻辑电路的框图





组合逻辑电路在电路结构上不包含存储单元，仅仅是由各种门电路组成，

## 4.2 组合逻辑电路的分析方法

---

已知逻辑电路  说明逻辑功能  
分析

分析方法步骤:

组合逻辑电路图  写出逻辑表达式  
 化简  列真值表  说明功能

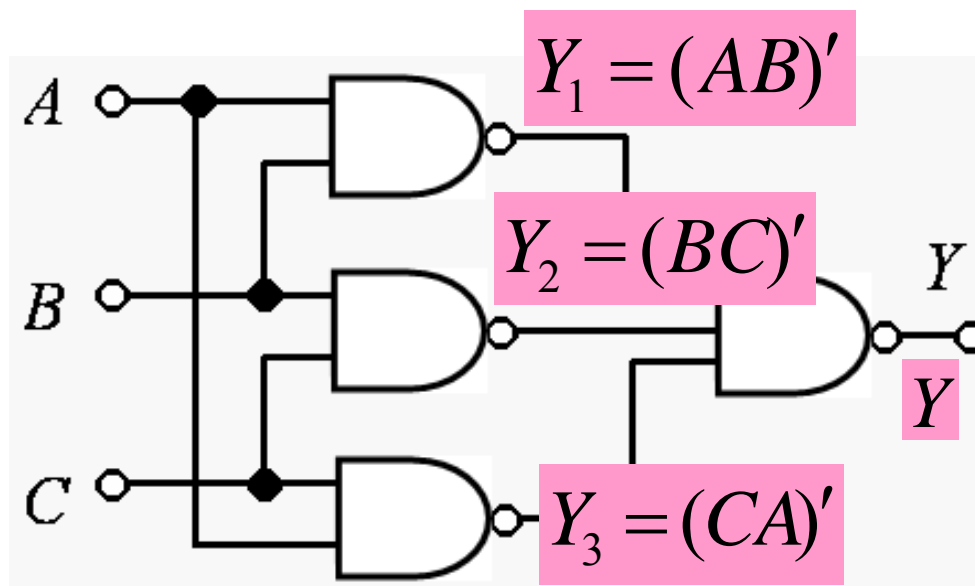
逻辑图

从输入到输出  
逐级写出

逻辑表  
达式

化简

最简与或  
表达式



1

$$Y = (Y_1 Y_2 Y_3)' = ((AB)'(BC)'(CA)')'$$

2

$$Y = AB + BC + CA$$

最简与或  
表达式

$$Y = AB + BC + CA$$

3

真值表

4

电路的逻  
辑功能

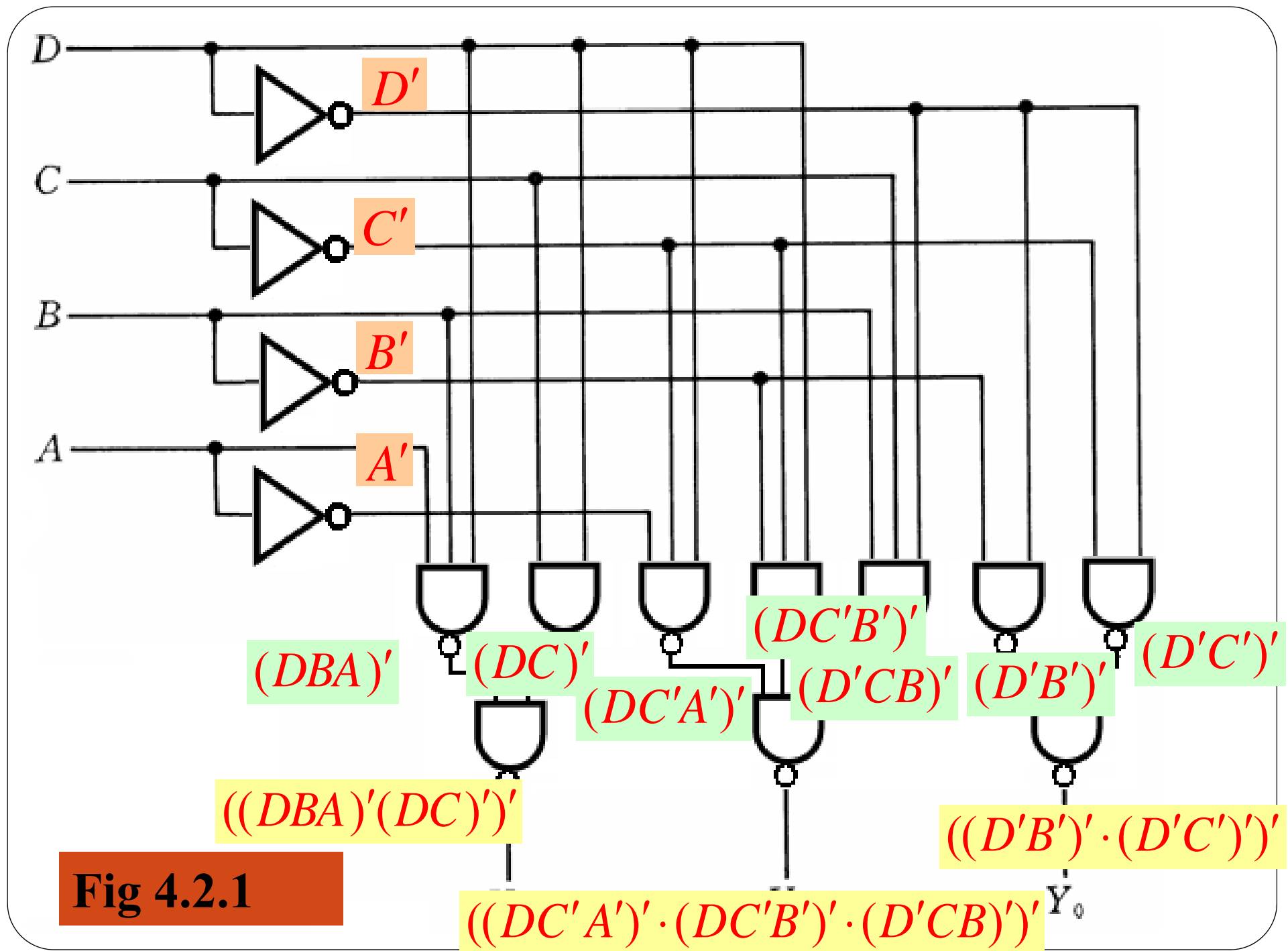
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

4

当输入A、B、C中有2个或3个为1时，输出Y为1，否则输出Y为0。

这个电路实际上是一种3人表决用的组合电路：只要有2票或3票同意，表决就通过。





**Fig 4.2.1**

解：

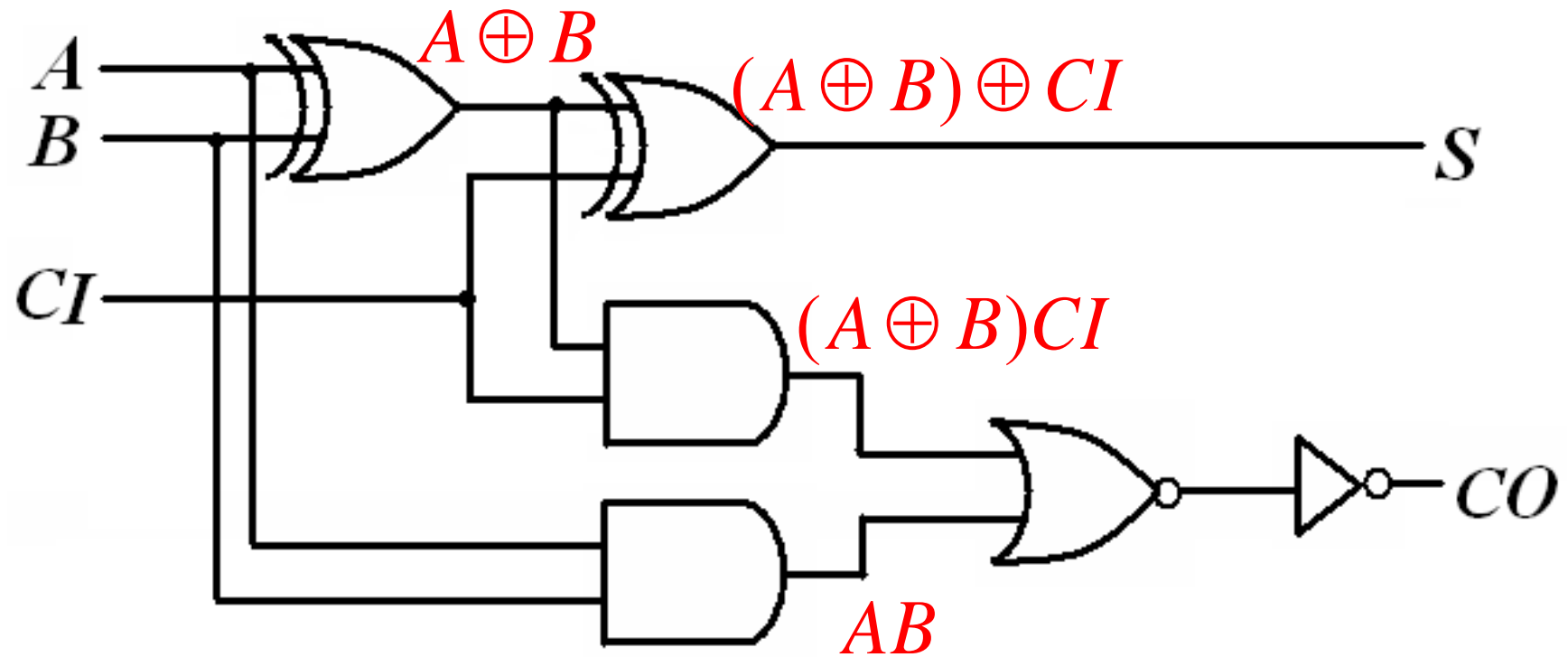
$$Y_2 = ((DBA)' \cdot (DC)')' = DBA + DC$$

$$Y_1 = ((DC'A')' \cdot (DC'B')' \cdot (D'CB)')' = DC'A' + DC'B' + D'CB$$

$$Y_0 = ((D'B')' \cdot (D'C')')' = D'B' + D'C'$$

D C B A	Y <sub>2</sub> Y <sub>1</sub> Y <sub>0</sub>	D C B A	Y <sub>2</sub> Y <sub>1</sub> Y <sub>0</sub>
0 0 0 0	0 0 1	1 0 0 0	0 1 0
0 0 0 1	0 0 1	1 0 0 1	0 1 0
0 0 1 0	0 0 1	1 0 1 0	0 1 0
0 0 1 1	0 0 1	1 0 1 1	1 0 0
0 1 0 0	0 0 1	1 1 0 0	1 0 0
0 1 0 1	0 0 1	1 1 0 1	1 0 0
0 1 1 0	0 1 0	1 1 1 0	1 0 0
0 1 1 1	0 1 0	1 1 1 1	1 0 0

由真值表知：该电路可用来判别输入的4位二进制数数值的范围。



**Fig 4.1.1**

$$S = A \oplus B \oplus CI$$

$$CO = (A \oplus B)CI + AB$$

$A$	$B$	$CI$	$S$	$CO$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = A \oplus B \oplus CI$$

$$CO = (A \oplus B)CI + AB$$

这是一个全  
加器电路

## 4.2 组合逻辑电路的设计方法

### 一、逻辑抽象

- 分析因果关系，确定输入/输出变量
- 定义逻辑状态的含意（赋值）
- 列出真值表

### 二、写出函数式

### 三、选定器件类型

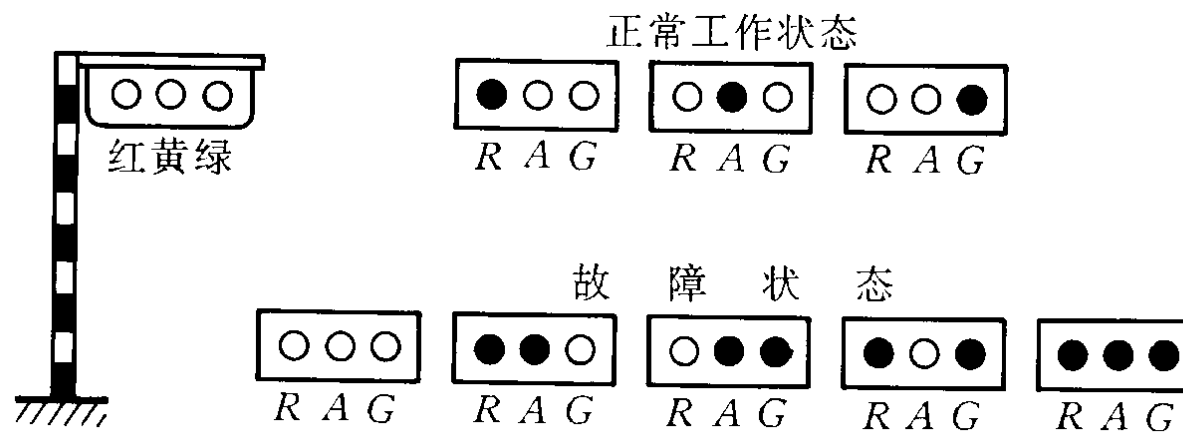
四、根据所选器件：对逻辑式化简（用门）  
变换（用MSI）  
或进行相应的描述（PLD）

### 五、画出逻辑电路图

### 六、工艺设计

# 设计举例：

- 设计一个监视交通信号灯状态的逻辑电路



## 设计举例：

### 1. 抽象

- 输入变量：

红 (Red)、黄 (Amber)、绿 (Green)

- 输出变量：

故障信号 (Z)

### 2. 写出逻辑表达式

$$Z = R'A'G' + R'AG + RA'G + RAG' + RAG$$

输入变量			输出
R	A	G	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



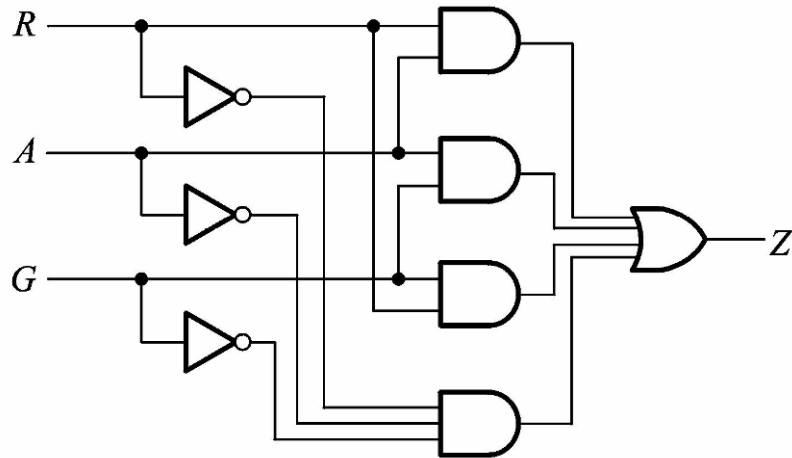
## 设计举例：

3. 选用小规模SSI器件

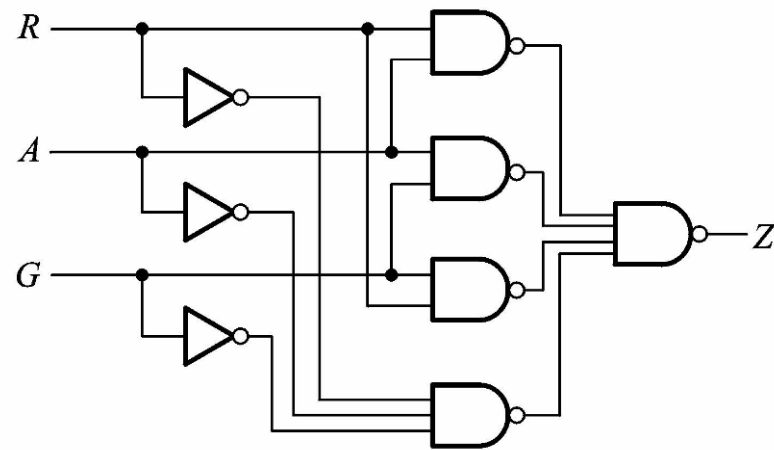
4. 化简

$$Z = R' A' G' + RA + RG + AG$$

5. 画出逻辑图



$R \backslash AG$				
	00	01	11	10
0	1	0	1	0
1	0	1	1	1



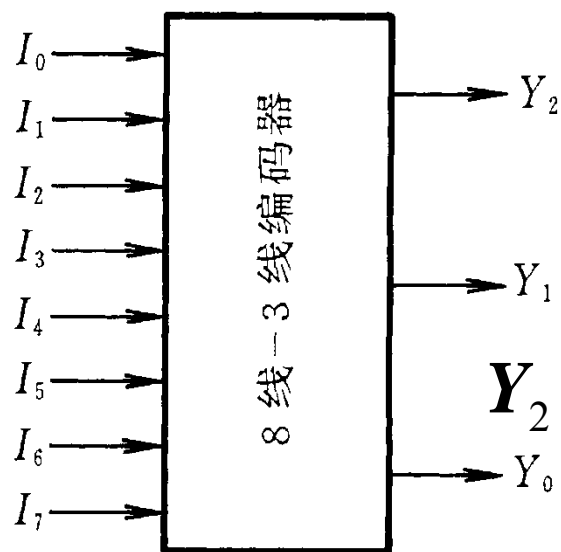
## 4.4 若干常用组合逻辑电路

### § 4.4.1 编码器

- 编码：将输入的每个高/低电平信号变成一个对应的二进制代码
- 普通编码器
- 优先编码器

# 一、普通编码器

- 特点：任何时刻只允许输入一个编码信号。
- 例：3位二进制普通编码器



输 入								输 出		
$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$Y_2$	$Y_1$	$Y_0$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

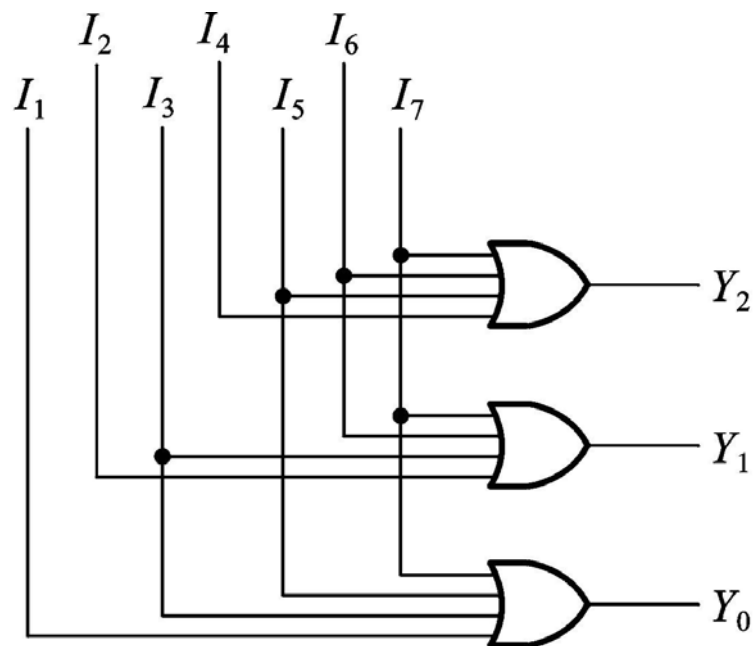
$$\begin{aligned}
 Y_2 = & I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0 \\
 & + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0'
 \end{aligned}$$

利用无关项化简，得：

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_0 = I_1 + I_3 + I_5 + I_7$$



## 二、优先编码器

- 特点：允许同时输入两个以上的编码信号，但只对其中优先权最高的一个进行编码。
- 例：8线-3线优先编码器
- （设 $I_7$ 优先权最高… $I_0$ 优先权最低）

输 入								输 出		
$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$Y_2$	$Y_1$	$Y_0$
X	X	X	X	X	X	X	1	1	1	1
X	X	X	X	X	X	1	0	1	1	0
X	X	X	X	X	1	0	0	1	0	1
X	X	X	X	1	0	0	0	1	0	0
X	X	X	1	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0
X	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0

$$Y_2 = I_7 + I_7' I_6 + I_7' I_6' I_5 + I_7' I_6' I_5' I_4$$

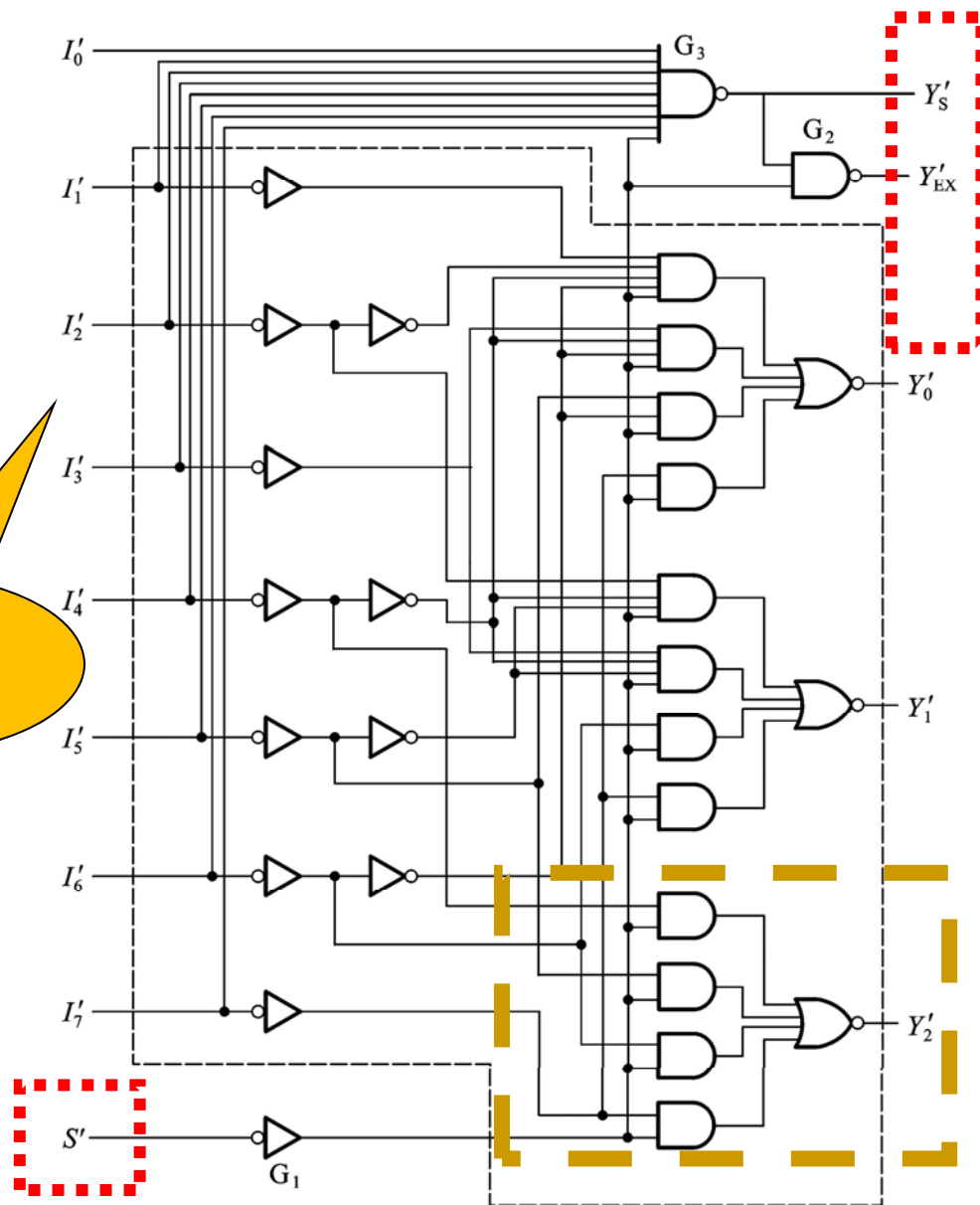


$$Y_2 = I_7 + I_6 + I_5 + I_4$$

$$A + A'B = A + B$$

# 实例： 74HC148

低电平



$$Y'_2 = (I_7 + I_6 + I_5 + I_4)'$$

选通信号

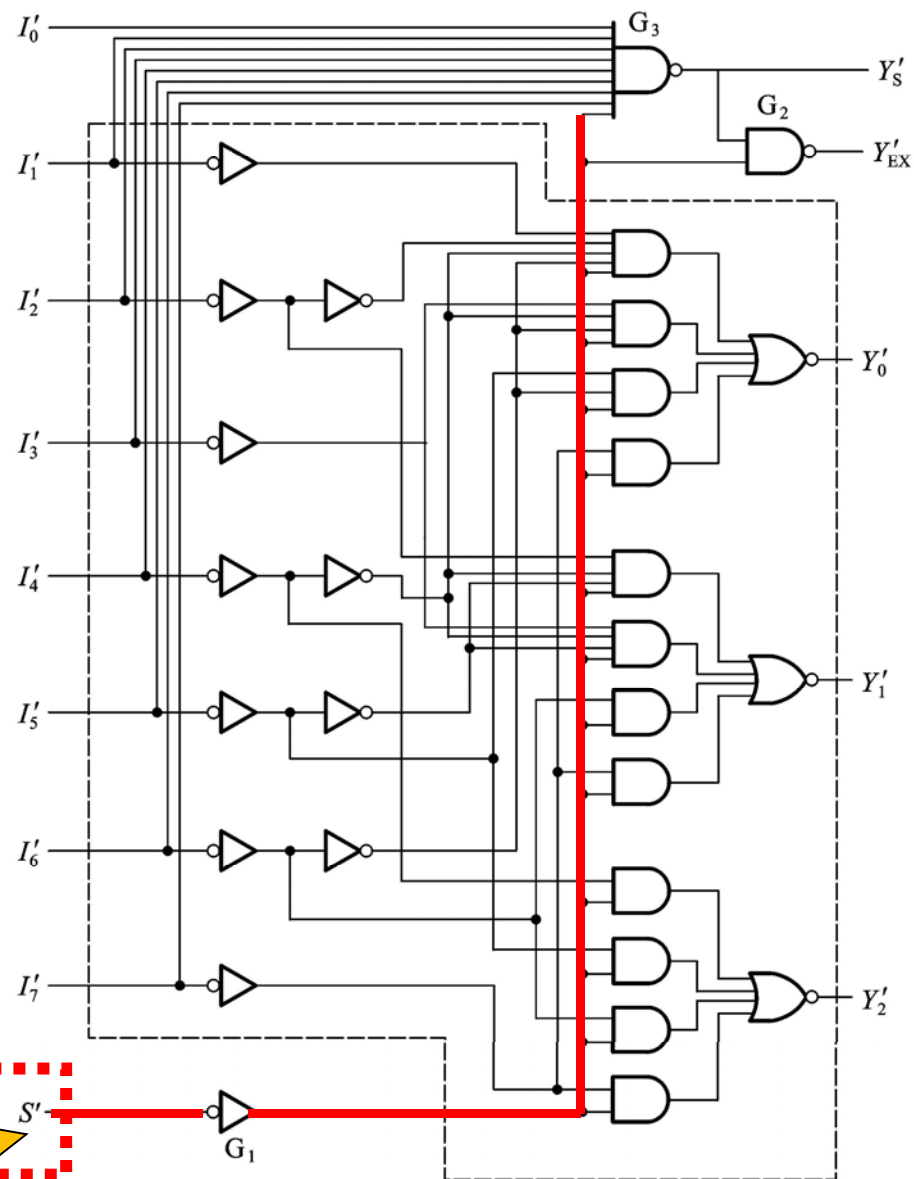
$$Y'_2 = [(I_7 + I_6 + I_5 + I_4)S']$$

$$Y'_2 = [(I_7 + I_6 + I_5 + I_4)S']$$

$$Y'_1 = [(I_7 + I_6 + I_3I'_4I'_5 + I_2I'_4I'_5)S']$$

$$Y'_0 = [(I_7 + I'_6I_5 + I_3I'_4I'_6 + I_1I_2I'_4I'_6)S']$$

S'为0时，电路正常工作  
S'为1时，输出锁定高

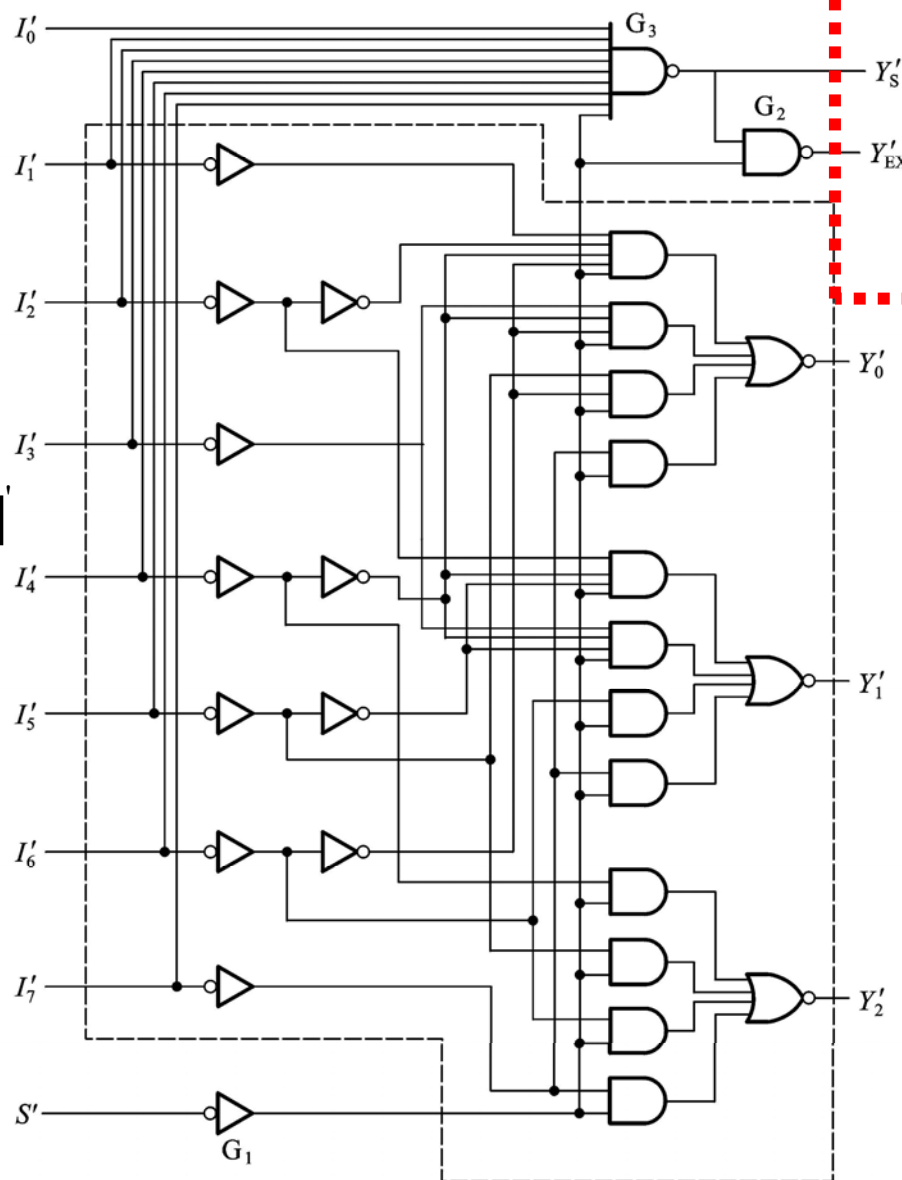


选通信号

为0时，电路工作  
无编码输入

$$\begin{aligned}
 Y'_S &= (I'_7 I'_6 I'_5 I'_4 I'_3 I'_2 I'_1 I'_0 S')' \\
 &= I_7 + I_6 + I_5 + I_4 + I_3 + I_2 + I_1 + I_0 + S' \\
 Y'_{EX} &= [(I'_7 I'_6 I'_5 I'_4 I'_3 I'_2 I'_1 I'_0 S')' \quad S'] \\
 &= [(I_7 + I_6 + I_5 + I_4 + I_3 + I_2 + I_1 + I_0) S']' \\
 &= I'_7 I'_6 I'_5 I'_4 I'_3 I'_2 I'_1 I'_0 + S'
 \end{aligned}$$

为0时，电路工作  
有编码输入



附加输出信号



[illegible]

## 附加输出信号的状态及含意

$Y'_S$	$Y'_{EX}$	状态
1	1	不工作
0	1	工作，但无输入
1	0	工作，且有输入
0	0	不可能出现

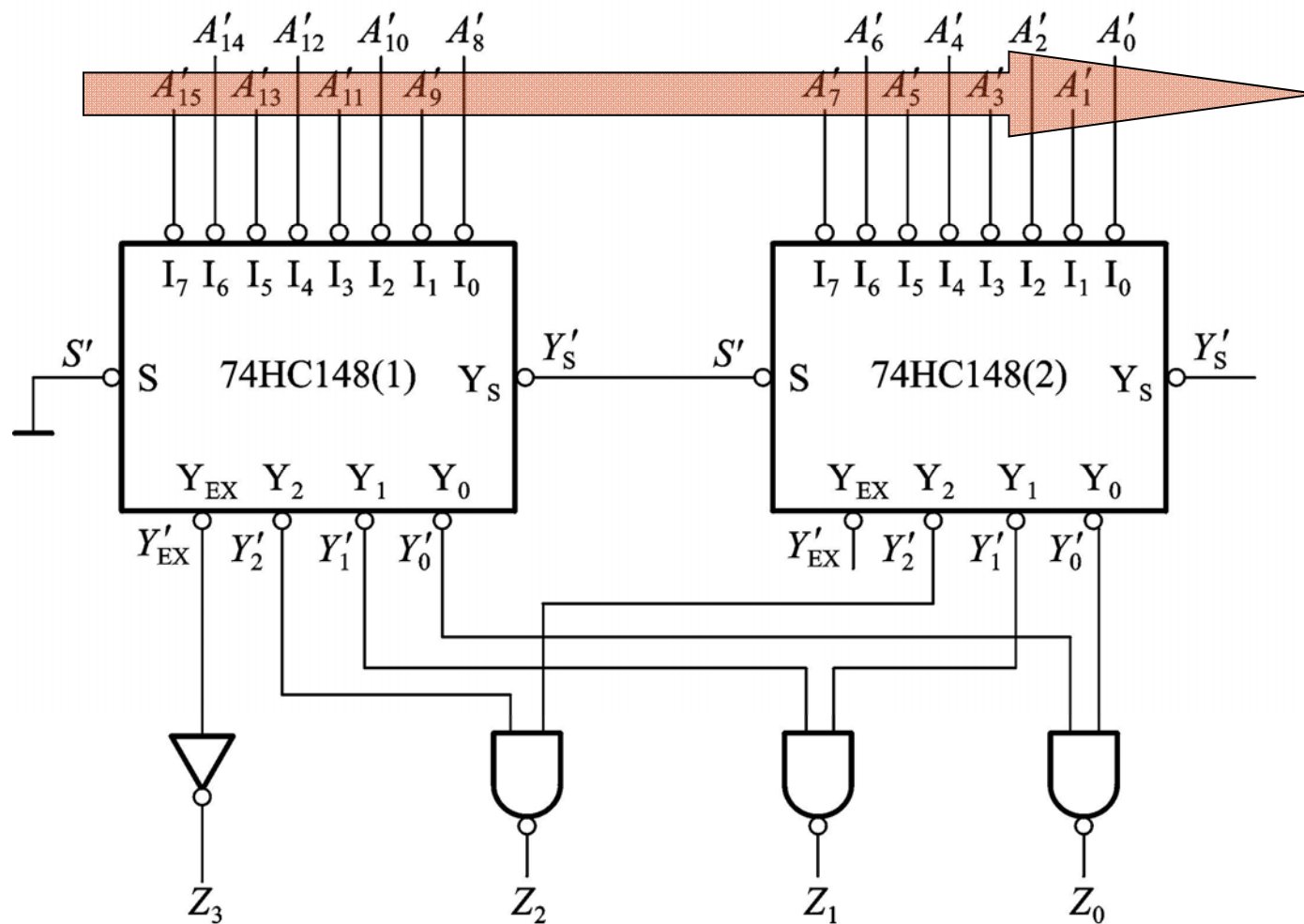
## 控制端扩展功能举例：

- 例：用两片8线-3线优先编码器

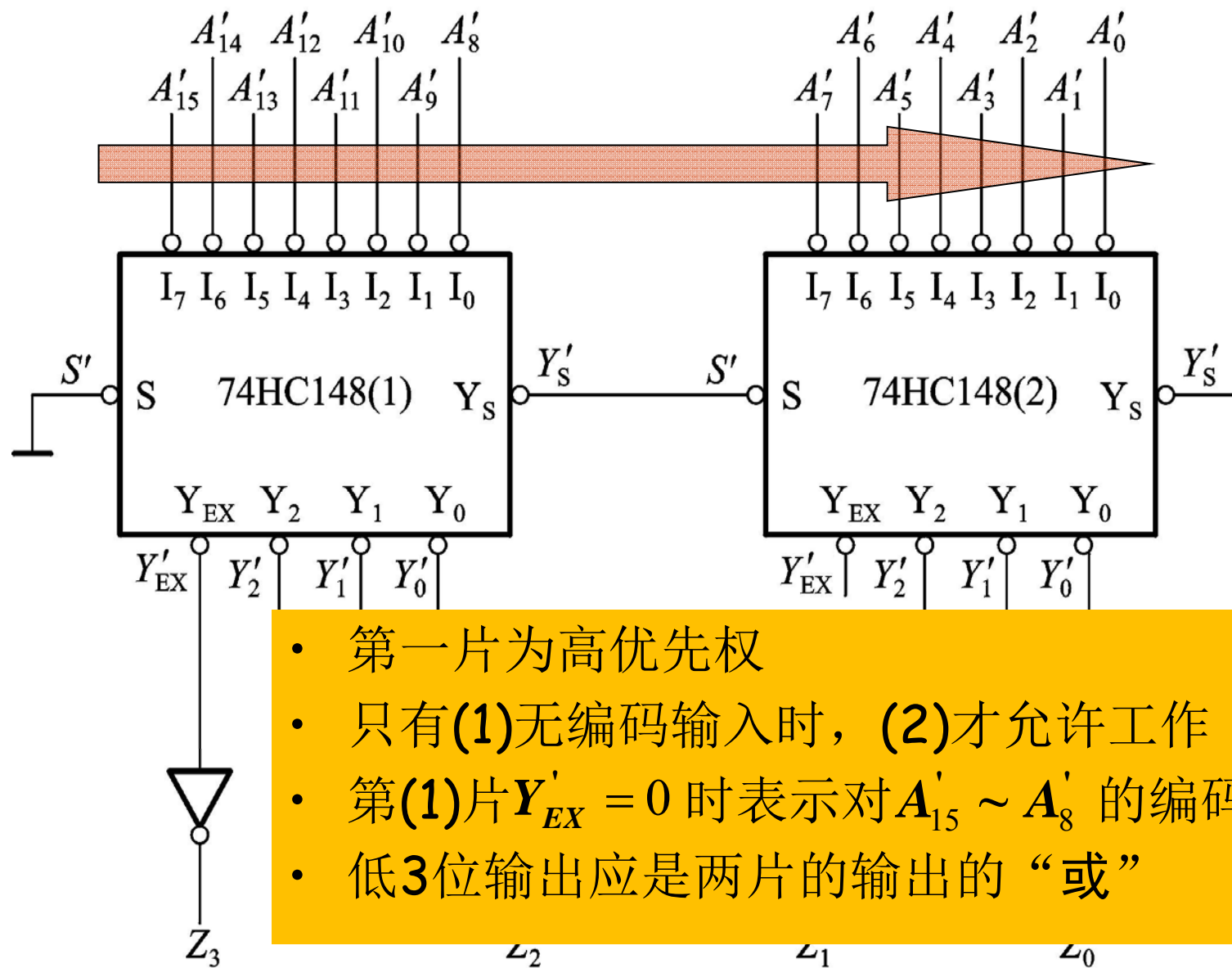


16线-4线优先编码器

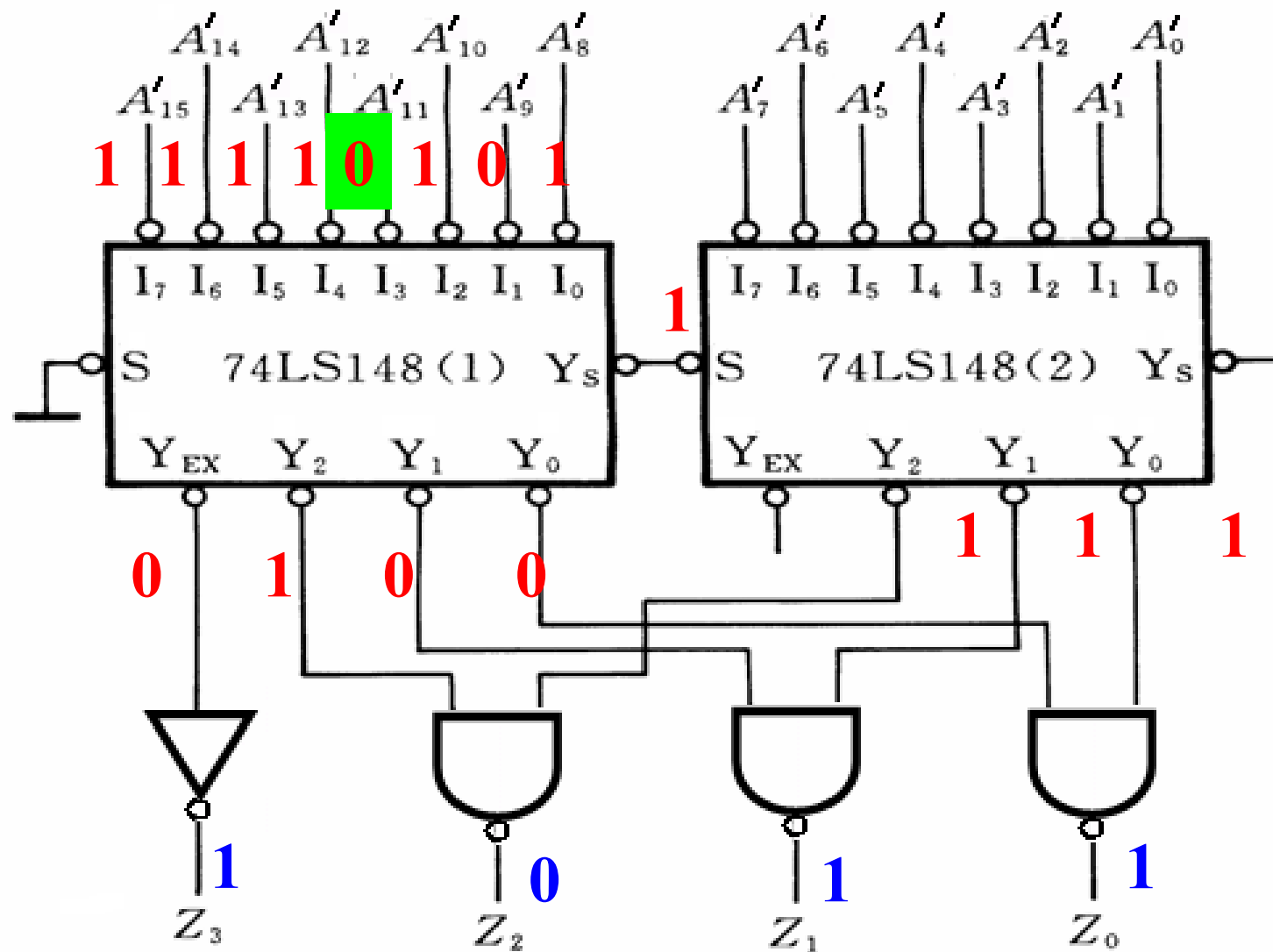
其中， $A'_{15}$  的优先权最高 • • •



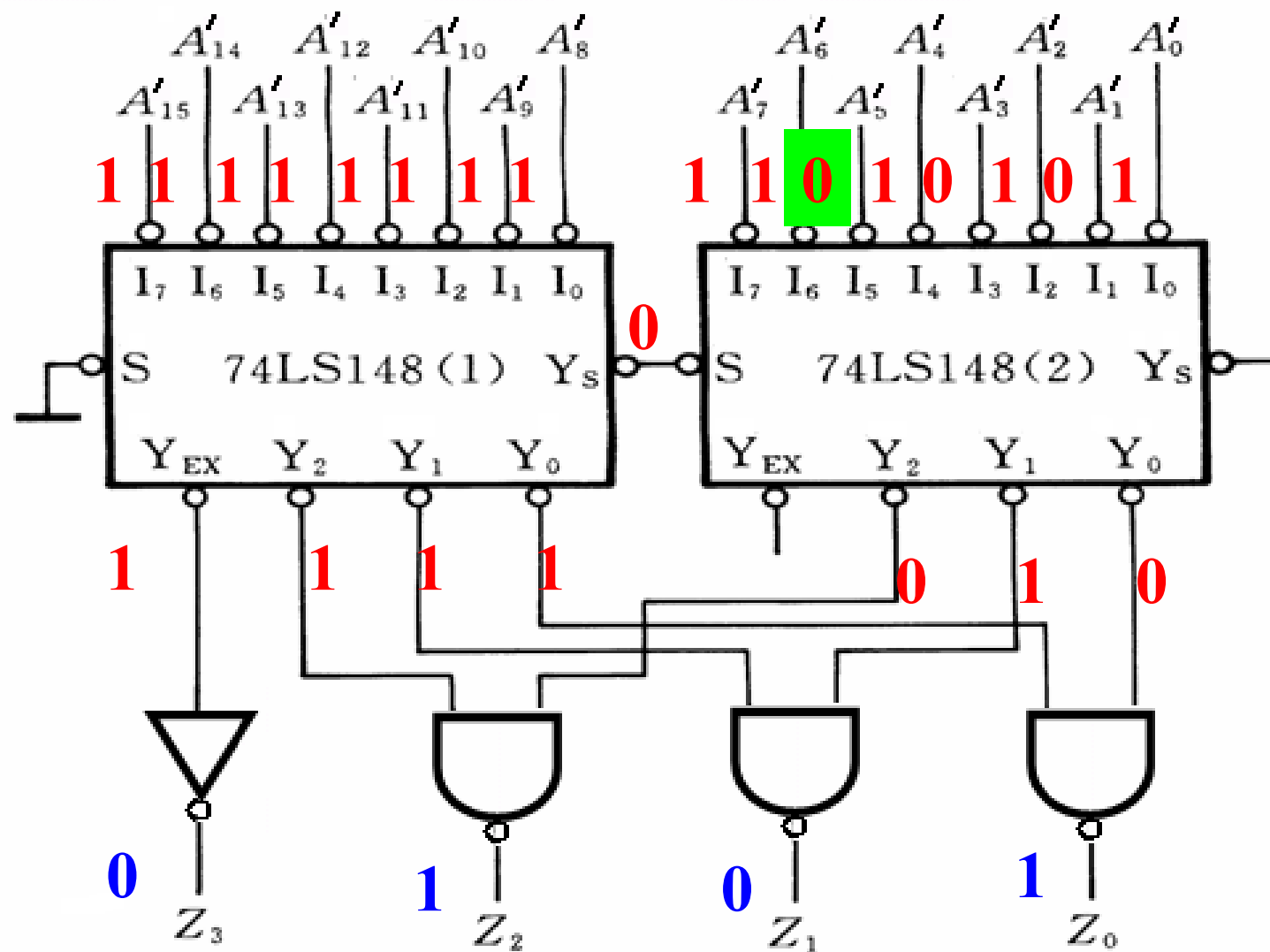
$Y'_S$	$Y'_{EX}$	状态
1	1	不工作
0	1	工作，但无输入
1	0	工作，且有输入
0	0	不可能出现



- 第一片为高优先权
- 只有(1)无编码输入时，(2)才允许工作
- 第(1)片  $Y'_{EX} = 0$  时表示对  $A'_{15} \sim A'_8$  的编码
- 低3位输出应是两片的输出的“或”



(1)片处于编码状态,(2)片被封锁。



(2)片处于编码状态

### 三、二-十进制优先编码器

- 将  $I'_9 \sim I'_1$  编成 0110  $\sim$  1110 (反码形式的BCD码)  
 $I'_9$   $I'_0$
- 的优先权最高, 最低 (Table 4.3.3)
- 输入的低电平信号变成一个对应的十进制的编码

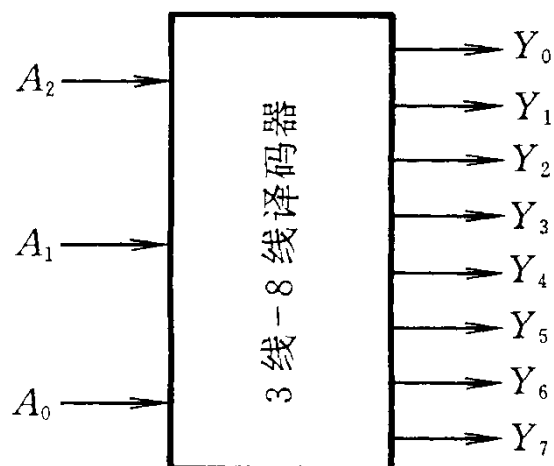


## § 4.4.2 译码器

- 译码：将每个输入的二进制代码译成对应的输出高、低电平信号。
- 常用的有：二进制译码器，二-十进制译码器，显示译码器等

### 一、二进制译码器

例：3线—8线译码器



输 入			输 出							
$A_2$	$A_1$	$A_0$	$Y_7$	$Y_6$	$Y_5$	$Y_4$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

真值表  
达式：



逻辑表

$$Y_0 = A_2' A_1' A_0' = m_0$$

$$Y_1 = A_2' A_1' A_0 = m_1$$



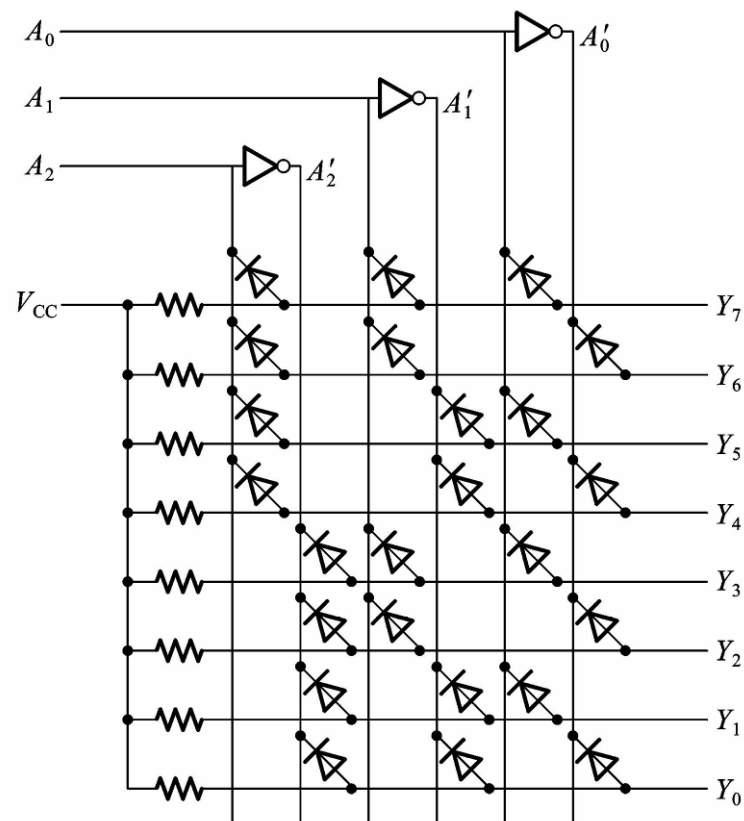
用电路进行实现

$$Y_2 = A_2' A_1 A_0' = m_2$$

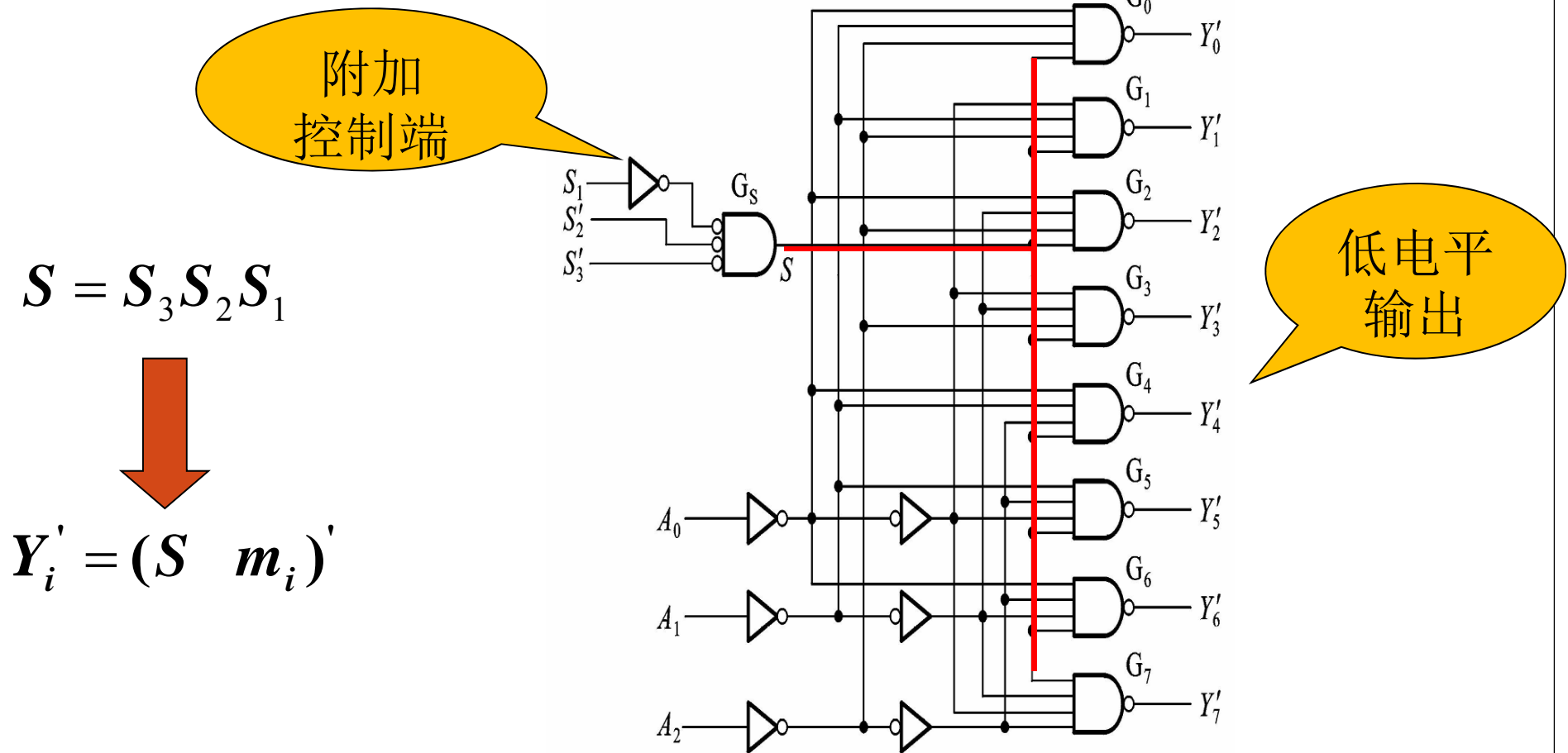
...

$$Y_7 = A_2 A_1 A_0 = m_7$$

用二极管与门阵列组成的3线—8线译码器



# 集成译码器实例：74HC138



## 74HC138的功能表：

输 入					输 出							
$S_1$	$S_2' + S_3'$	$A_2$	$A_1$	$A_0$	$Y_7'$	$Y_6'$	$Y_5'$	$Y_4'$	$Y_3'$	$Y_2'$	$Y_1'$	$Y_0'$
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

当 $S_1=1, S_2=0, S_3=0$  (即 $S=1$ ) 时, 可得输出

$$Y'_0 = (A'_2 A'_1 A'_0)' = m'_0$$

$$Y'_4 = (A_2 A'_1 A'_0)' = m'_4$$

$$Y'_1 = (A'_2 A'_1 A_0)' = m'_1$$

$$Y'_5 = (A_2 A'_1 A_0)' = m'_5$$

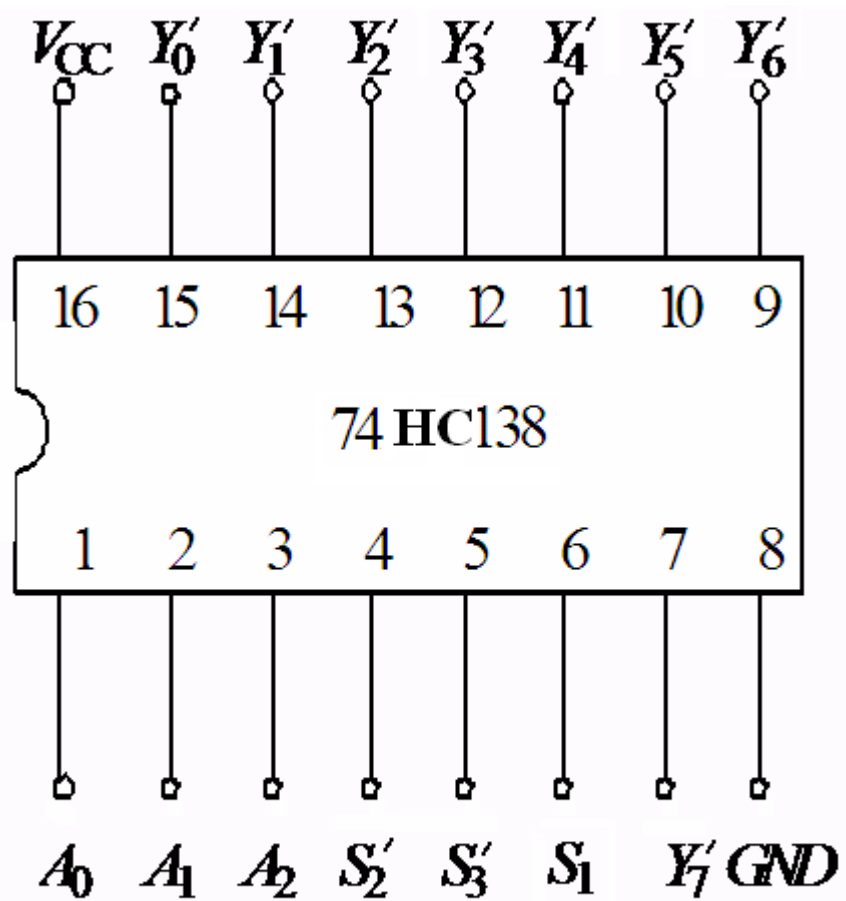
$$Y'_2 = (A'_2 A_1 A'_0)' = m'_2$$

$$Y'_6 = (A_2 A_1 A'_0)' = m'_6$$

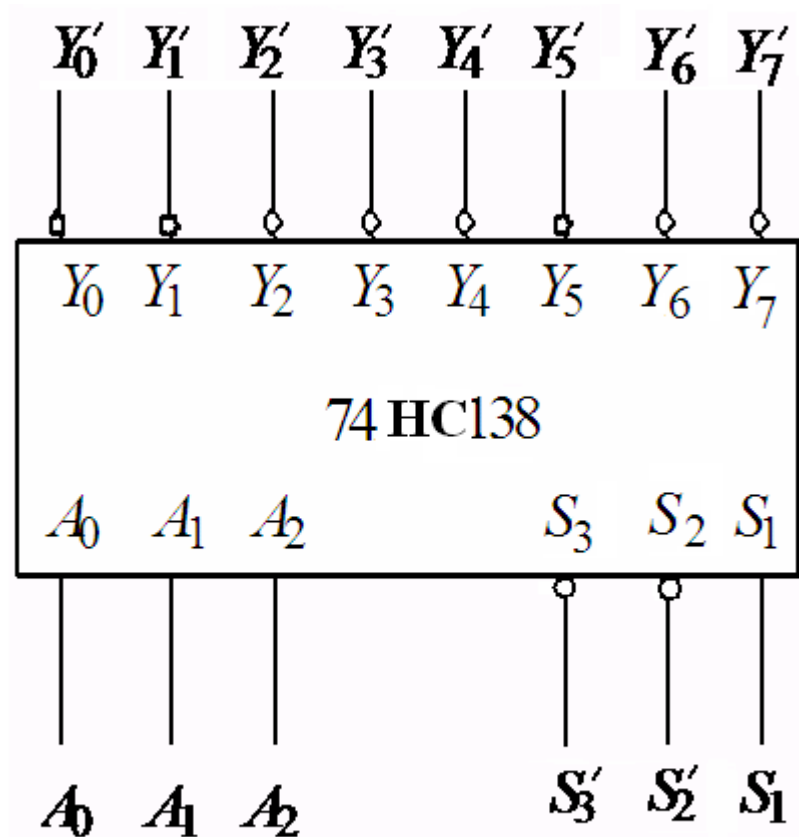
$$Y'_3 = (A'_2 A_1 A_0)' = m'_3$$

$$Y'_7 = (A_2 A_1 A_0)' = m'_7$$



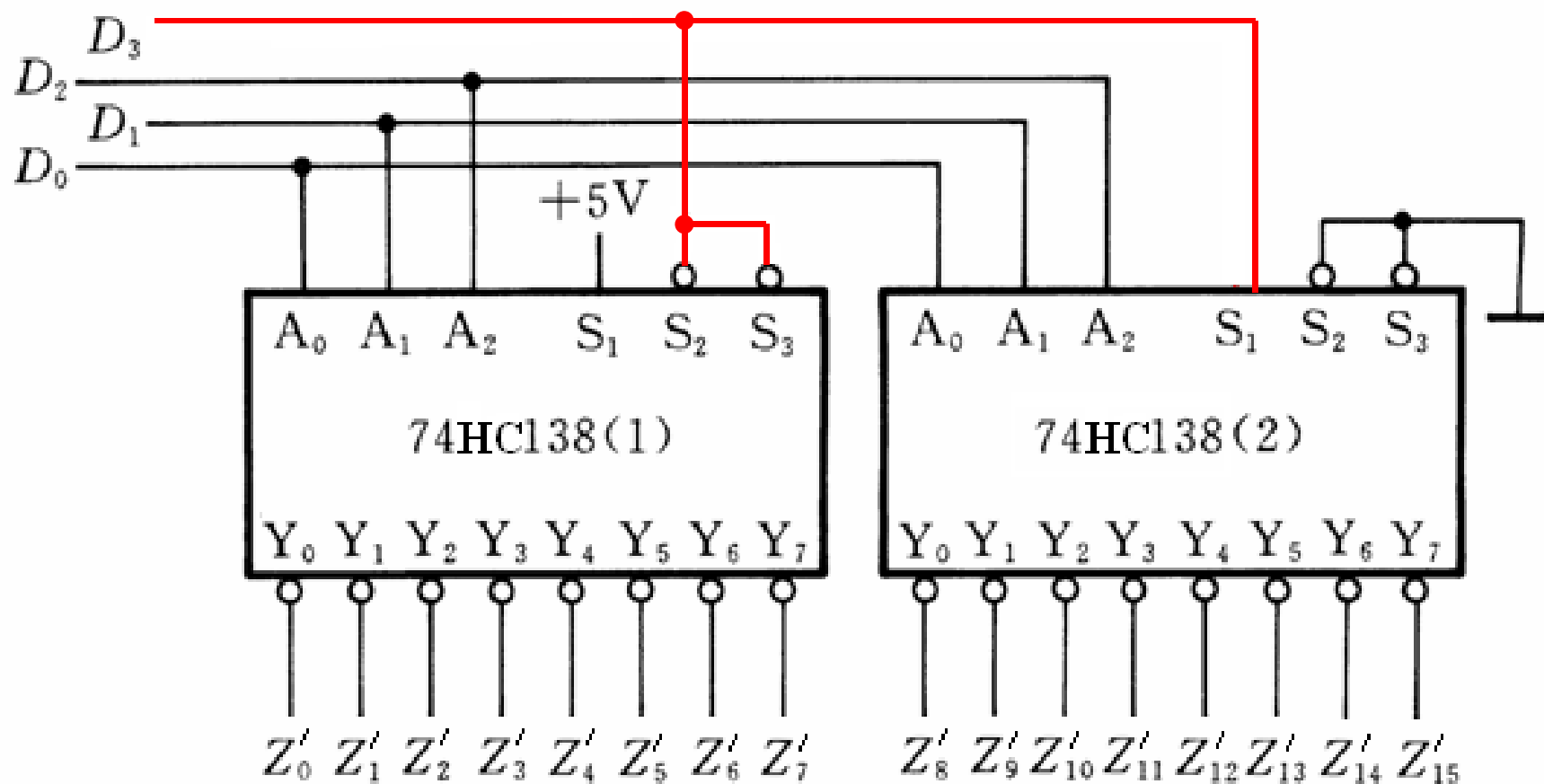


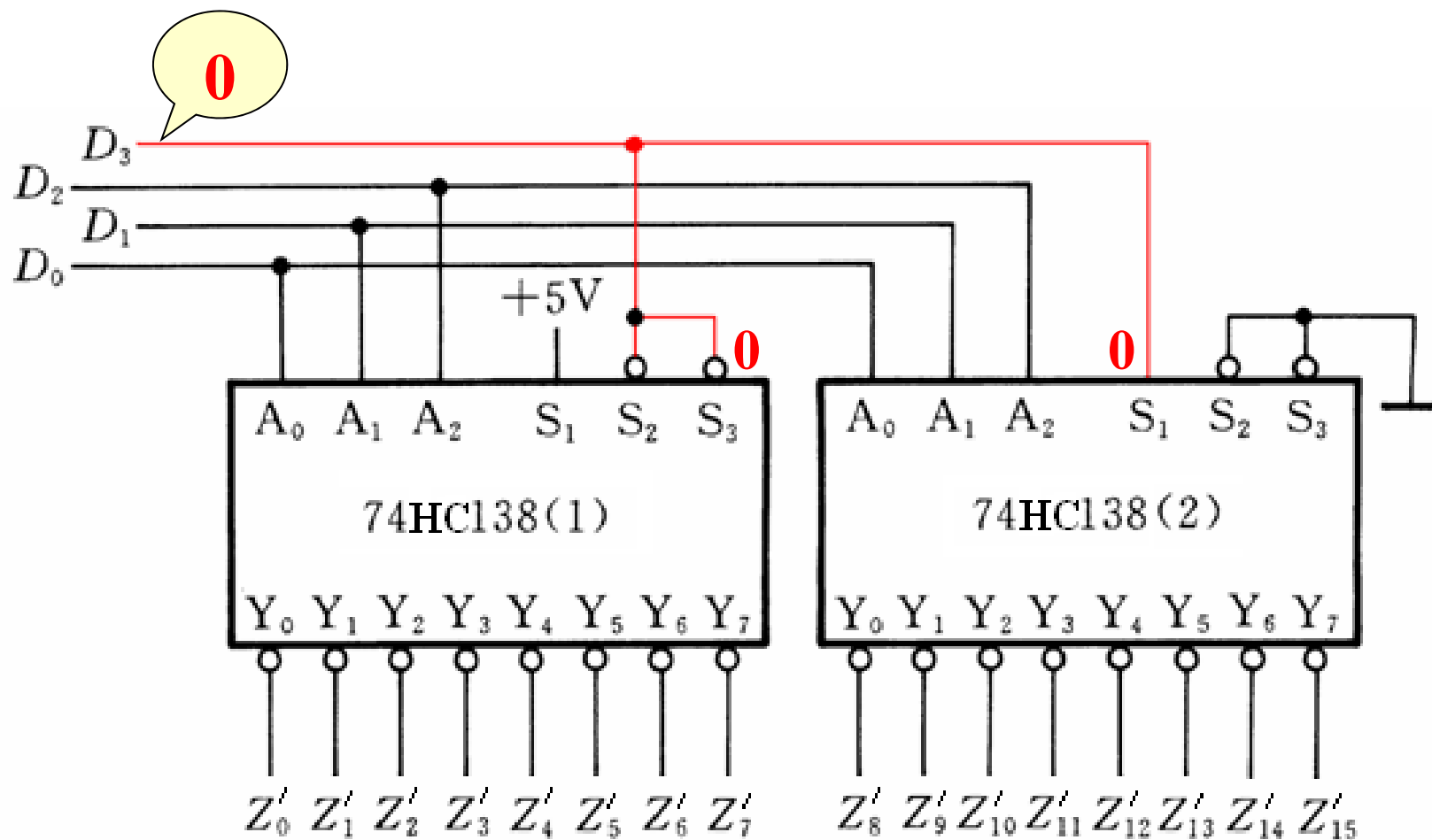
(a) 引脚排列图



(b) 逻辑功能示意图

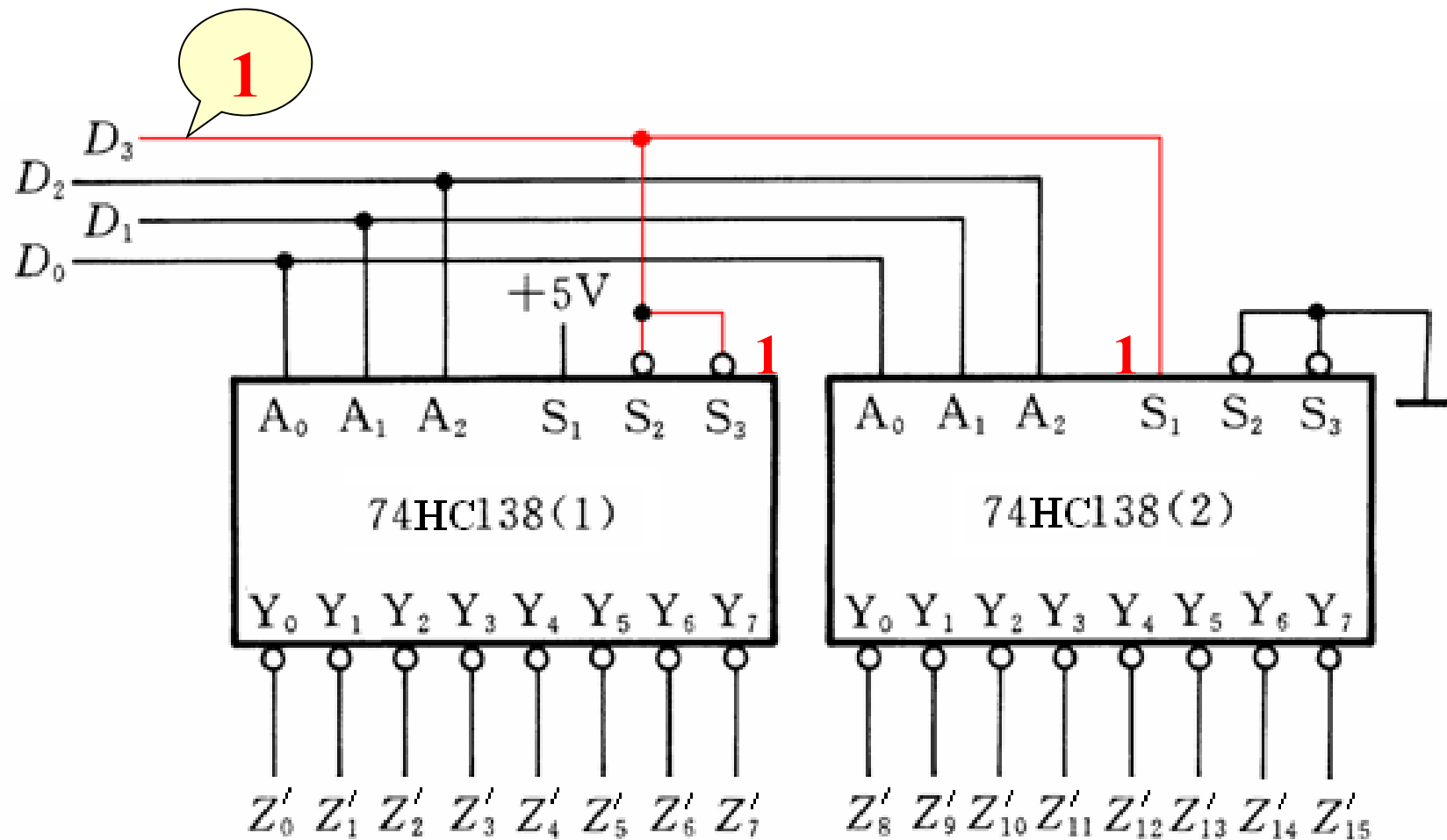
**例4.3.2：** 试用两片3线—8线译码器74HC138组成4线—16线译码器。





(1)片工作，(2)片禁止。若输入 $D_3D_2D_1D_0=0100$ 时，译码器 (1) 输出 11110111。





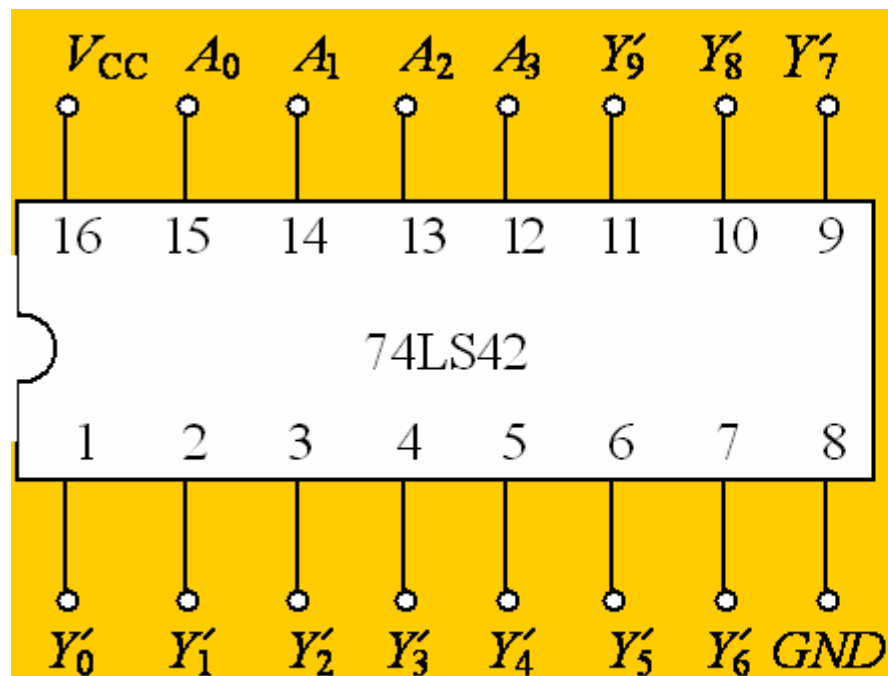
(2)片工作，(1)片禁止。若输入 $D_3D_2D_1D_0=1101$ 时，译码器(2)输出1111011。

## 二、二-十进制译码器

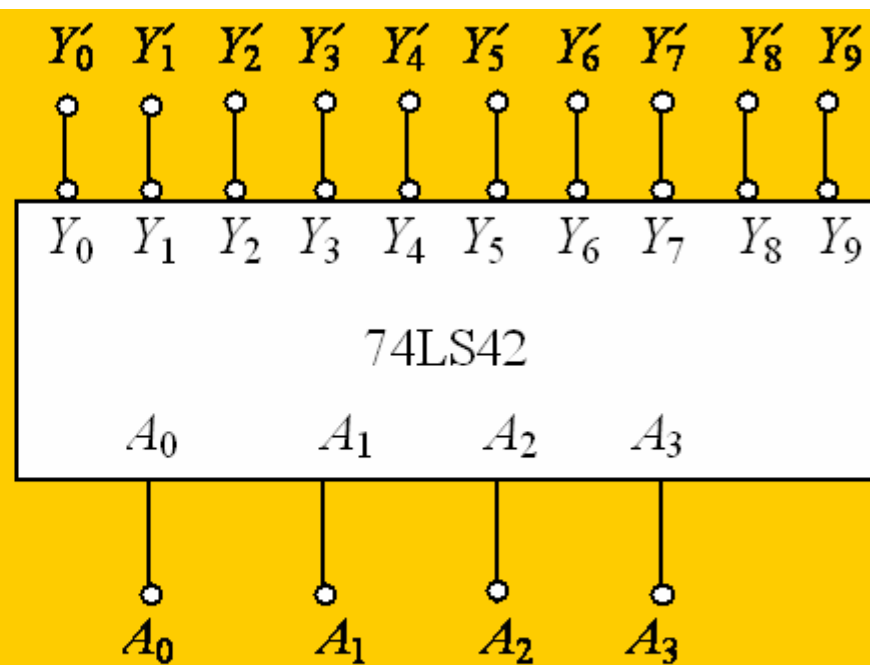
输入端：4      输出端：10

二-十进制译码器的输入是十进制数的4位二进制编码（BCD码），分别用 $A_3$ 、 $A_2$ 、 $A_1$ 、 $A_0$ 表示；输出的是与10个十进制数字相对应的10个信号，用 $Y_9 \sim Y_0$ 表示。由于二-十进制译码器有4根输入线，10根输出线，所以又称为**4线-10线译码器**。

## 集成8421 BCD码译码器74LS42



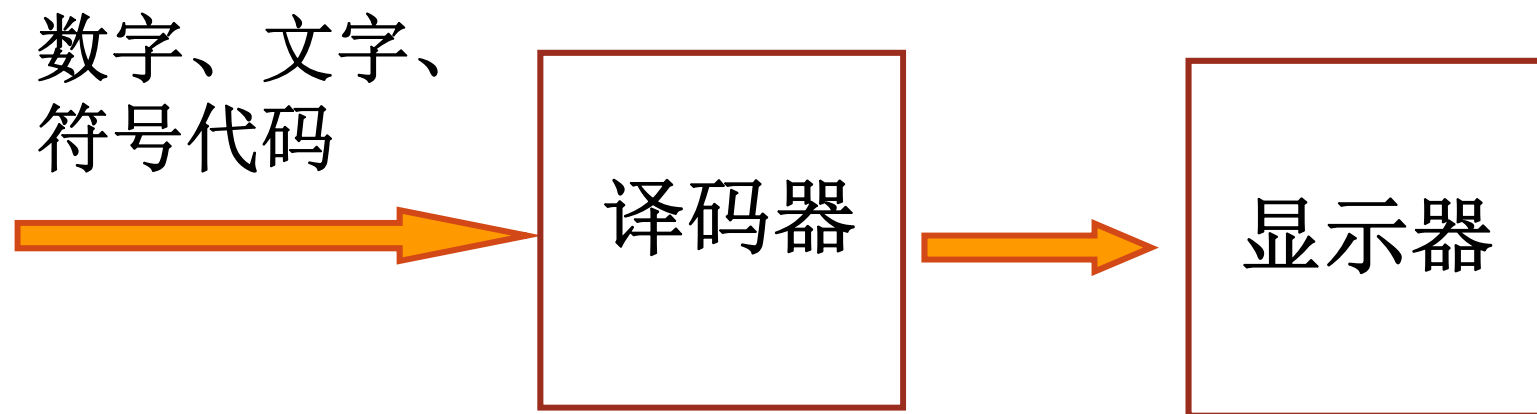
(a) 引脚排列图



(b) 逻辑功能示意图

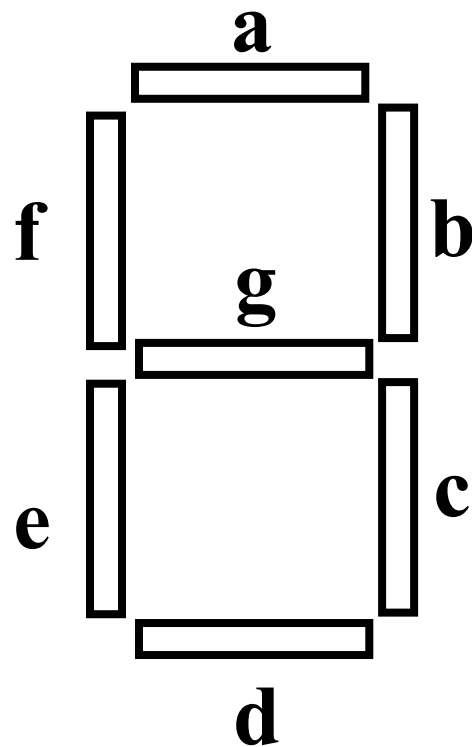
### 三、显示译码器

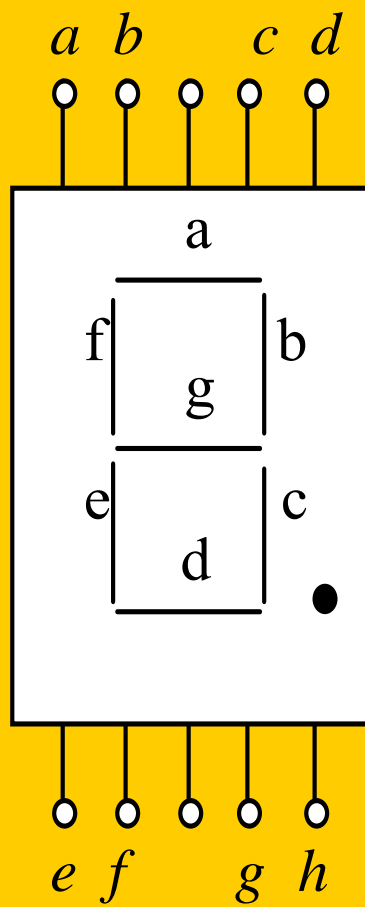
用来驱动各种显示器件，从而将用二进制代码表示的数字、文字、符号翻译成人们习惯的形式直观地显示出来的电路，称为显示译码器。



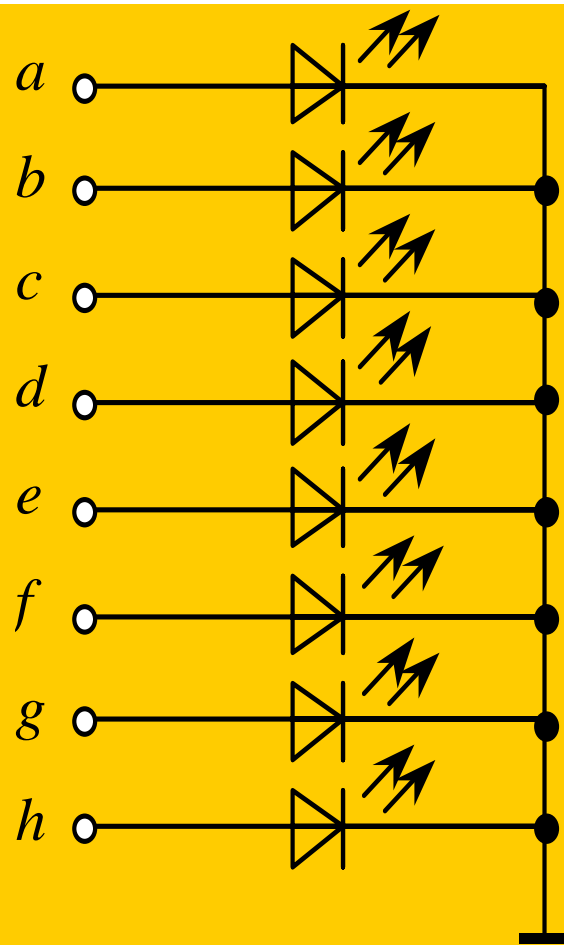
# 半导体数码管

显示器件： 常用的是七段显示器件

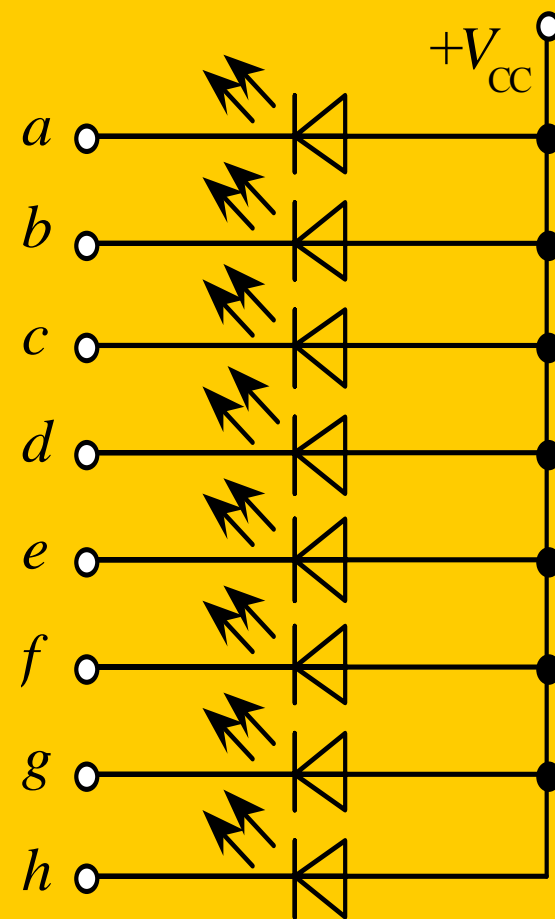




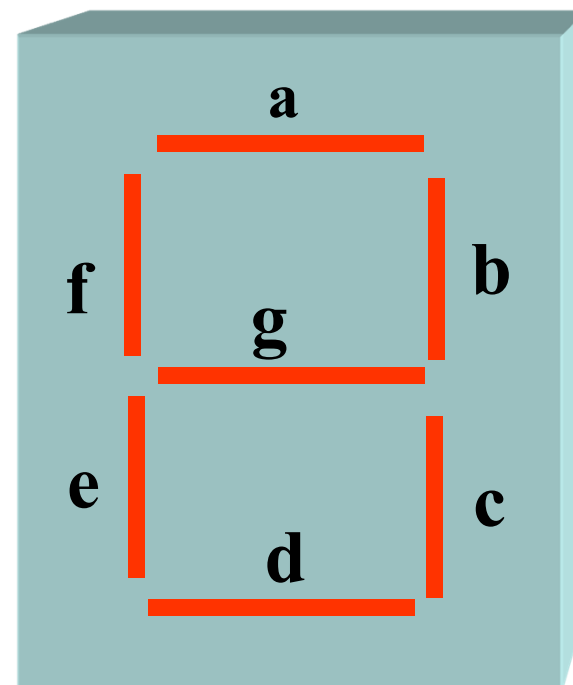
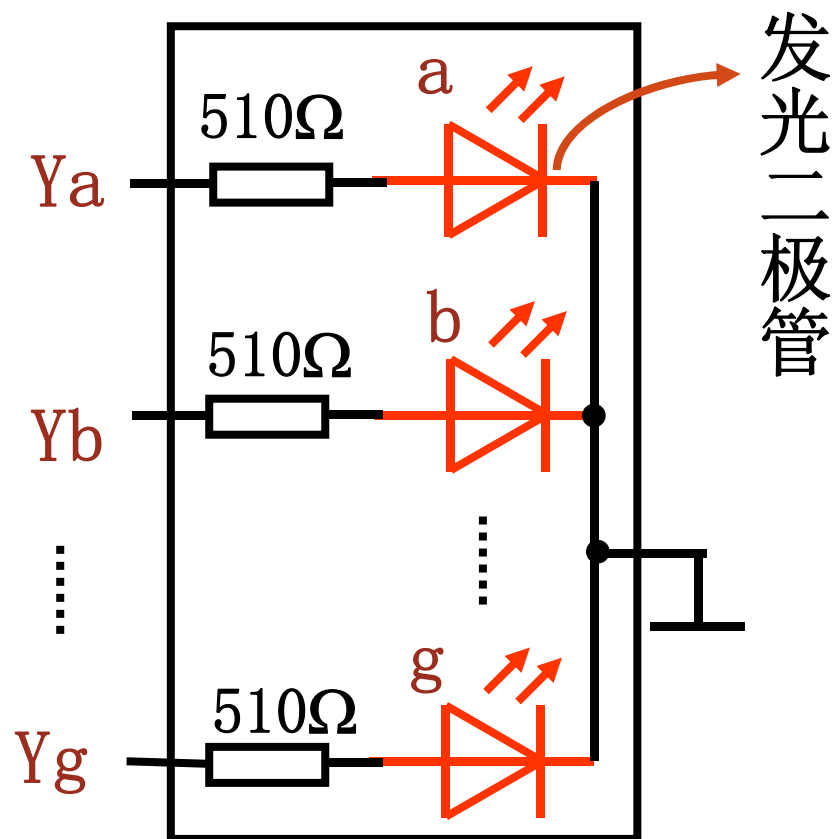
(a) 外形图



(b) 共阴极



(c) 共阳极



Ya-Yg: 控制信号

高电平时, 对应的LED亮

低电平时, 对应的LED灭

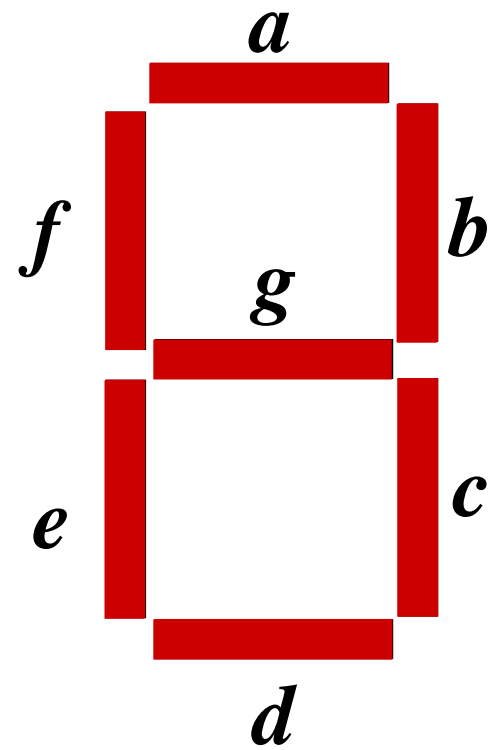
*a* *b* *c* *d* *e* *f* *g*

1 1 1 1 1 1 0

0 1 1 0 0 0 0

1 1 0 1 1 0 1

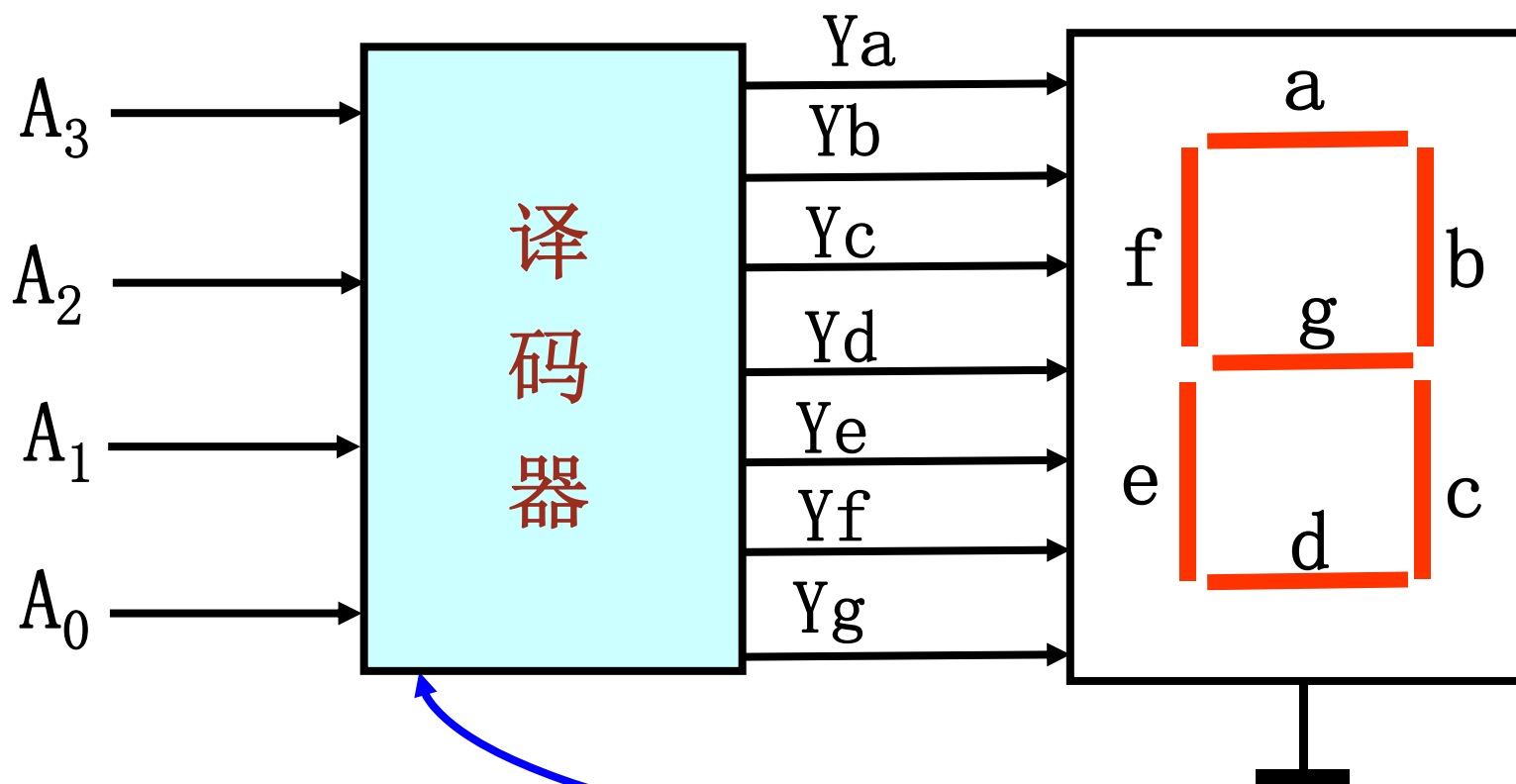
⋮





## BCD—七段显示译码器

$A_3-A_0$ : 输入数据



要设计的七段显示译码器

十进制数	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	Y <sub>a</sub>	Y <sub>b</sub>	Y <sub>c</sub>	Y <sub>d</sub>	Y <sub>e</sub>	Y <sub>f</sub>	Y <sub>g</sub>	显示字形
0	0 0 0 0	1	1	1	1	1	1	0	0
1	0 0 0 1	0	1	1	0	0	0	0	1
2	0 0 1 0	1	1	0	1	1	0	1	2
3	0 0 1 1	1	1	1	1	0	0	1	3
4	0 1 0 0	0	1	1	0	0	1	1	4
5	0 1 0 1	1	0	1	1	0	1	1	5
6	0 1 1 0	0	0	1	1	1	1	1	6
7	0 1 1 1	1	1	1	0	0	0	0	7
8	1 0 0 0	1	1	1	1	1	1	1	8
9	1 0 0 1	1	1	1	0	0	1	1	9

先设计输出Y<sub>a</sub>的逻辑表示式及电路图

Ya

A<sub>1</sub>A<sub>0</sub>

A<sub>3</sub>A<sub>2</sub>

	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	0	1	0	0
10	1	1	0	0

$$Y_a = (A'_3 A'_2 A'_1 A_0 + A_3 A_1 + A_2 A'_0)'$$

## 七段显示译码器7448引脚排列图

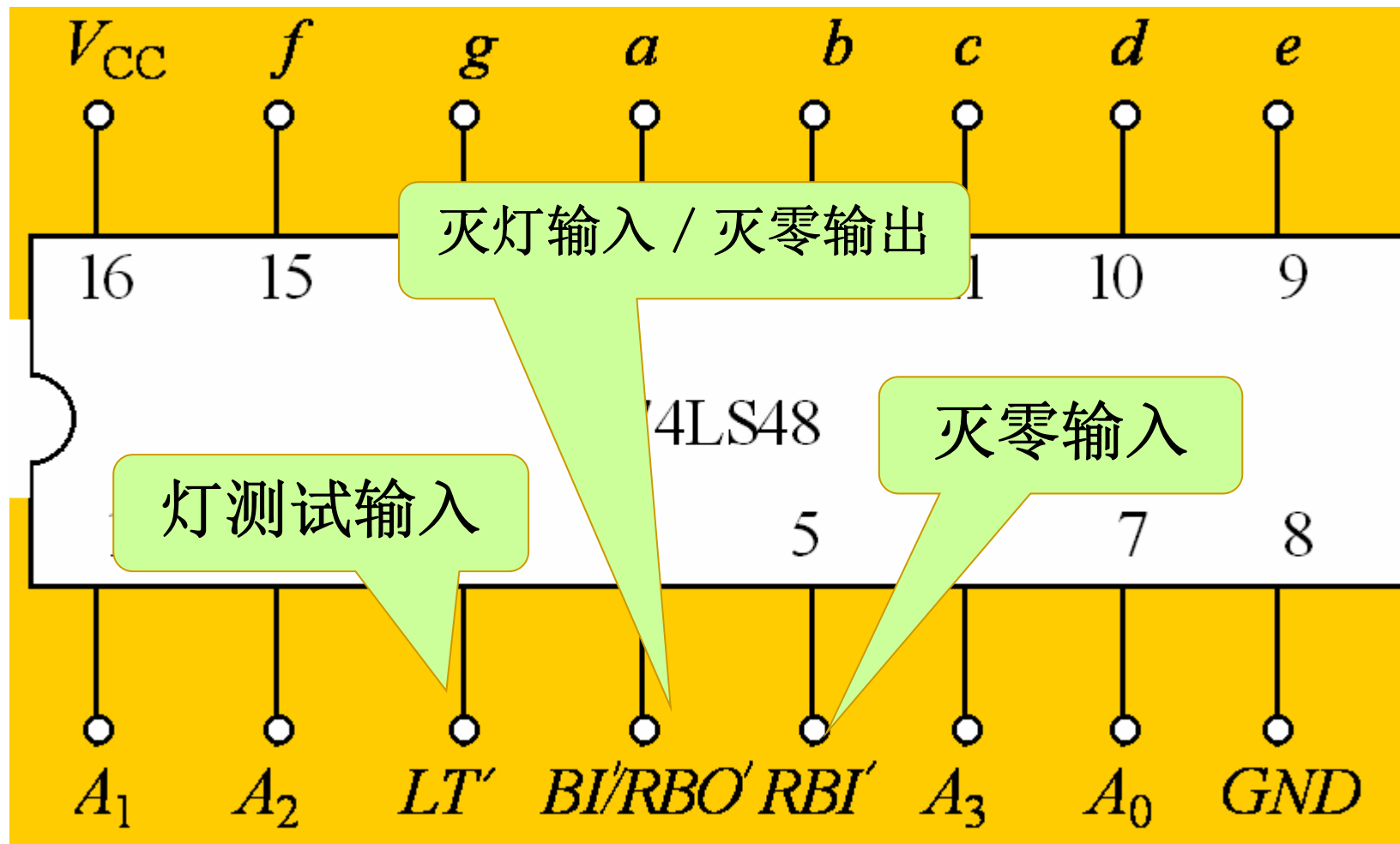
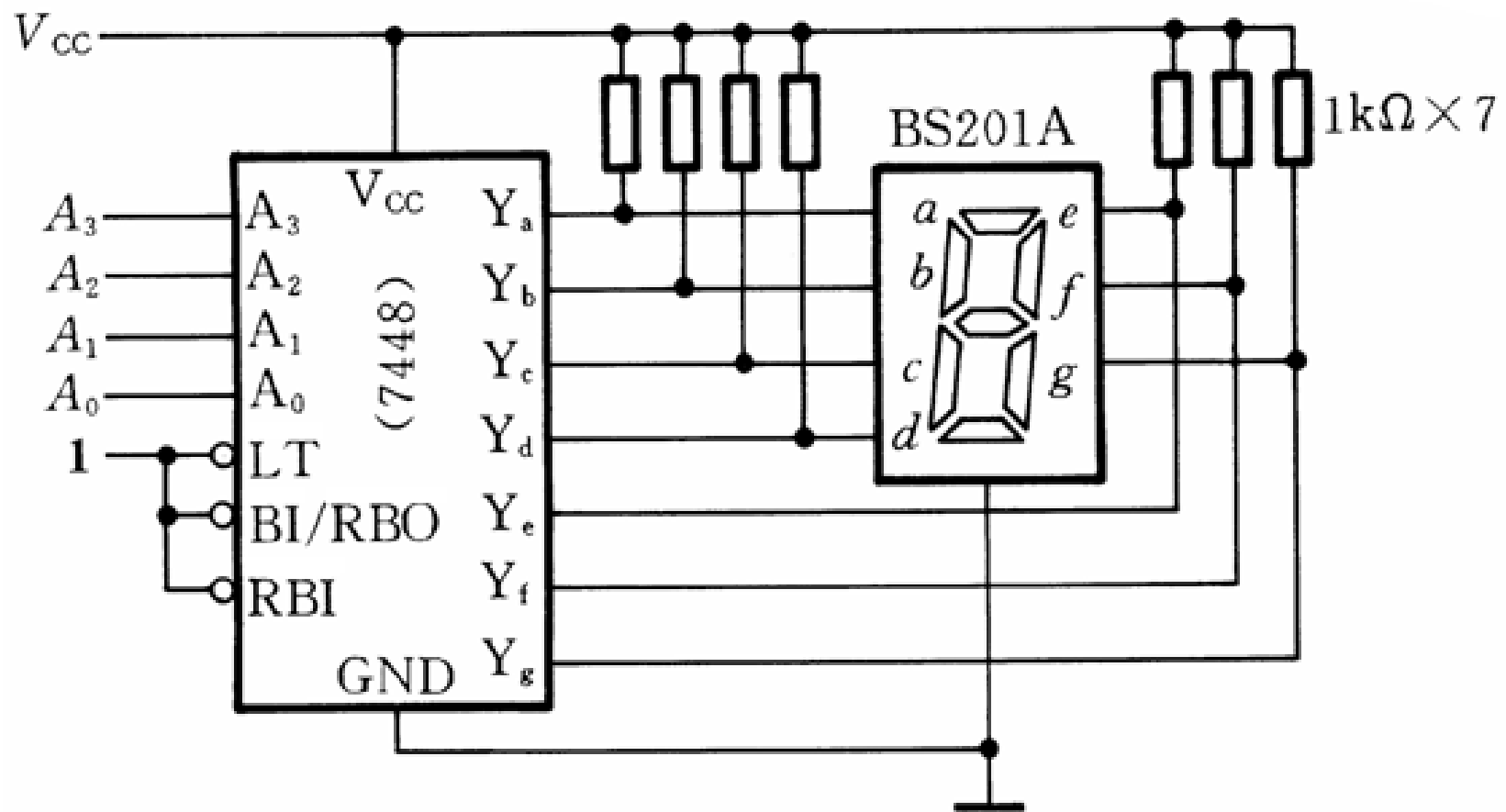


图4.3.18 用7448驱动BS201的连接方法



RBI和RBO配合使用，可使多位数字显示时的最高位及小数点后最低位的0不显示

			6	7	.	9		
--	--	--	---	---	---	---	--	--

## 四、译码器的应用

**例4.3.3：**试用3线—8线译码器74HC138设计一个多输出的组合逻辑电路。输出逻辑函数式为

$$\begin{cases} Z_1 = AC' + A'BC + AB'C \\ Z_2 = BC + A'B'C \\ Z_3 = A'B + AB'C \\ Z_4 = A'BC' + B'C' + ABC \end{cases}$$

解：        化为最小项之和的形式：

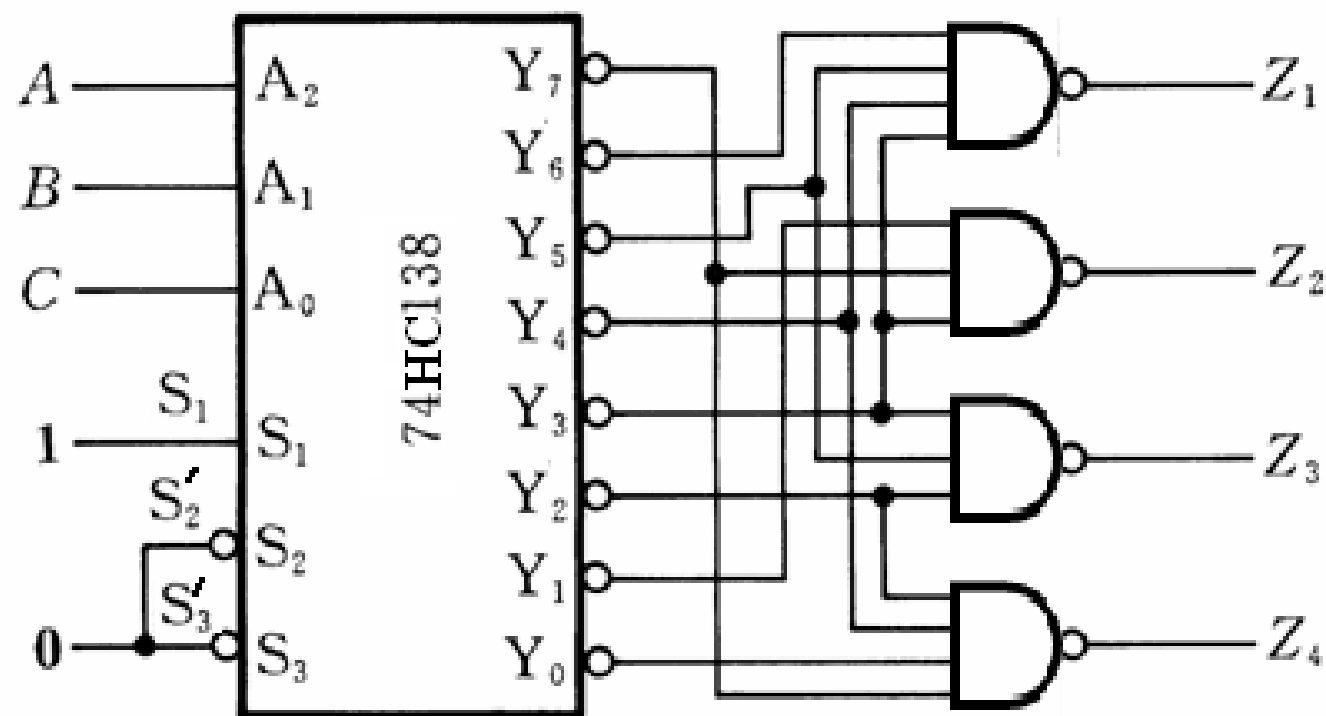
$$\left\{ \begin{array}{l} Z_1 = ABC' + AB'C' + A'BC + AB'C \\ Z_2 = ABC + A'BC + A'B'C \\ Z_3 = A'BC + A'BC' + AB'C \\ Z_4 = A'BC' + AB'C' + A'B'C' + ABC \end{array} \right.$$



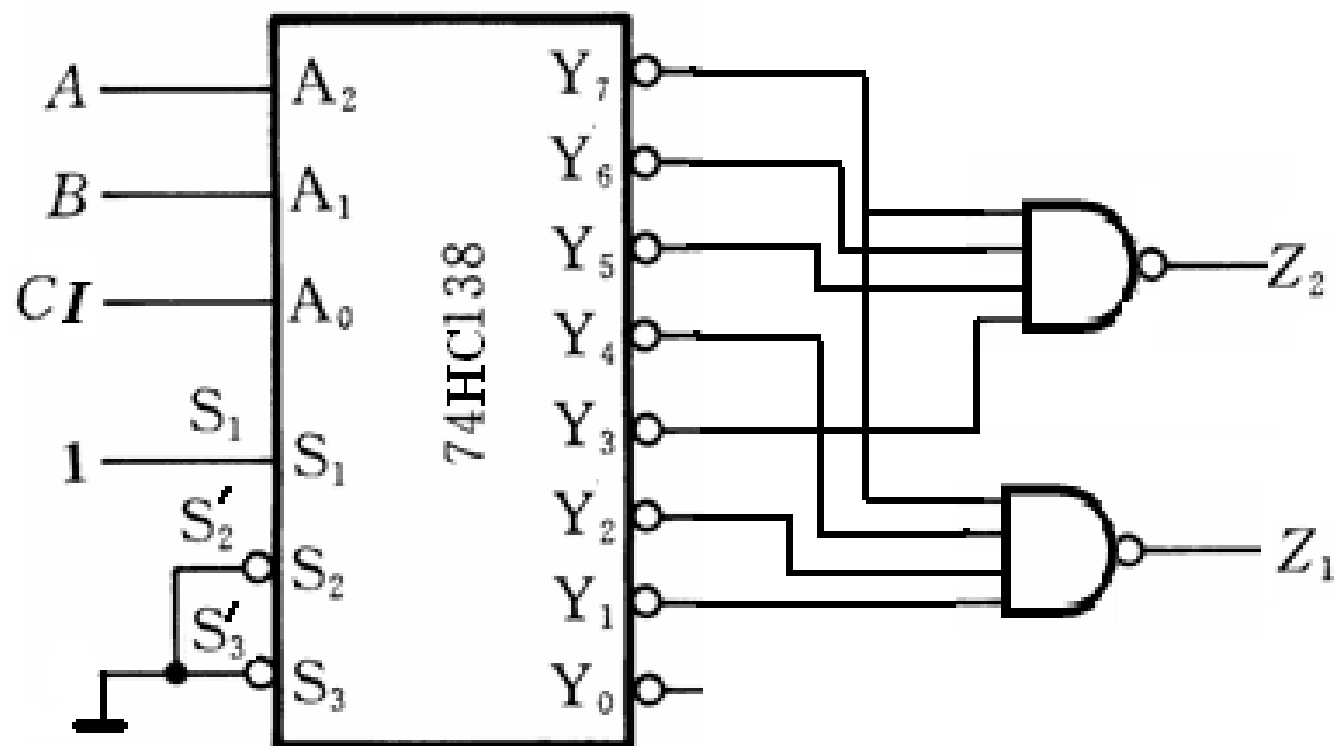
当 $S_1=1, S_2'=S_3'=0$ 时,令 $A_2=A, A_1=B, A_0=C$ ,则

$$\begin{cases} Z_1 = m_3 + m_4 + m_5 + m_6 = (m'_3 \cdot m'_4 \cdot m'_5 \cdot m'_6)' \\ Z_2 = m_1 + m_3 + m_7 = (m'_1 \cdot m'_3 \cdot m'_7)' \\ Z_3 = m_2 + m_3 + m_7 = (m'_2 \cdot m'_3 \cdot m'_7)' \\ Z_4 = m_0 + m_2 + m_4 + m_7 = (m'_0 \cdot m'_2 \cdot m'_4 \cdot m'_7)' \end{cases}$$

## 画电路图



例：分析下图电路逻辑功能。



解:  $Z_1 = (Y'_1 \cdot Y'_2 \cdot Y'_4 \cdot Y'_7)'$

$$= ((A'_2 A'_1 A_0)' \cdot (A'_2 A_1 A'_0)' \cdot (A_2 A'_1 A'_0)' \cdot (A_2 A_1 A_0)')'$$

$$= A'_2 A'_1 A_0 + A'_2 A_1 A'_0 + A_2 A'_1 A'_0 + A_2 A_1 A_0$$

$$= A'B'CI + A'BCI' + AB'CI' + ABCI$$

$$Z_2 = (Y'_3 \cdot Y'_5 \cdot Y'_6 \cdot Y'_7)'$$

$$= ((A'_2 A_1 A_0)' \cdot (A_2 A'_1 A_0)' \cdot (A_2 A_1 A'_0)' \cdot (A_2 A_1 A_0)')'$$

$$= A'_2 A_1 A_0 + A_2 A'_1 A_0 + A_2 A_1 A'_0 + A_2 A_1 A_0$$

$$= A'BCI + AB'CI + ABCI' + ABCI$$

$A$	$B$	$CI$	$Z_1$	$Z_2$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Z_1 = A'B'CI + A'BCI' + AB'CI' + ABCI$$

$$Z_2 = A'BCI + AB'CI + ABCI' + ABCI$$

这是一个全加器电路

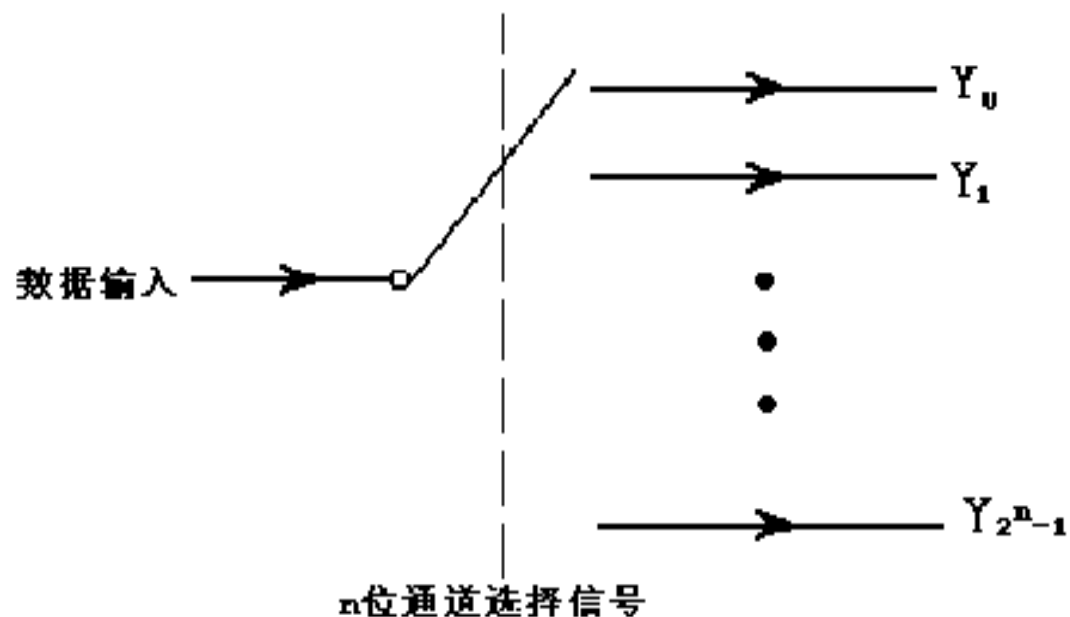
# 数据分配器

**定义：** 将公共数据线上的信号根据需要送到多个不同通道上去的逻辑电路。

输入端:1个

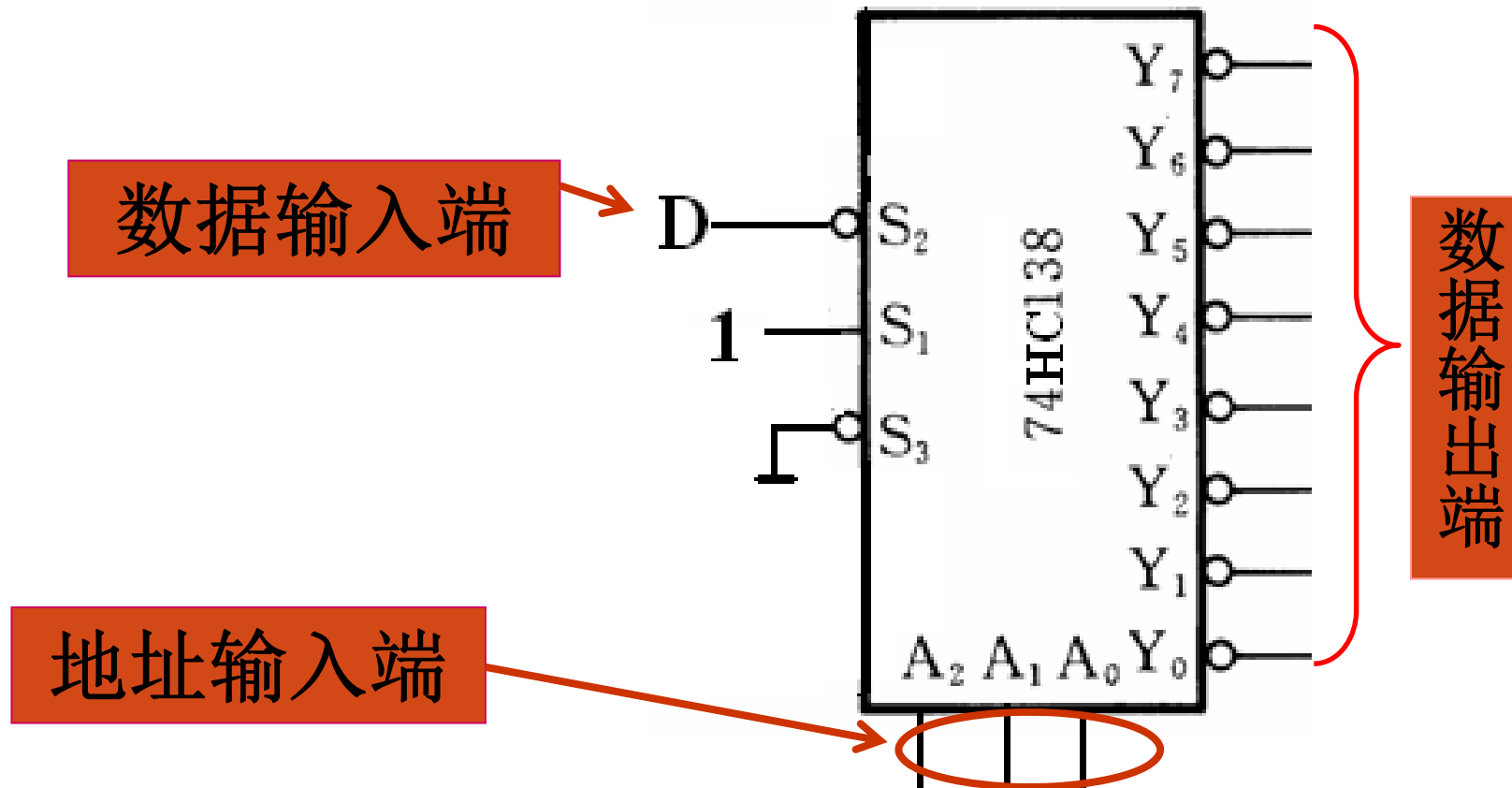
输出端: $2^n$ 个

框图:



数据分配器示意图

## 由74HC138构成的1路-8路数据分配器

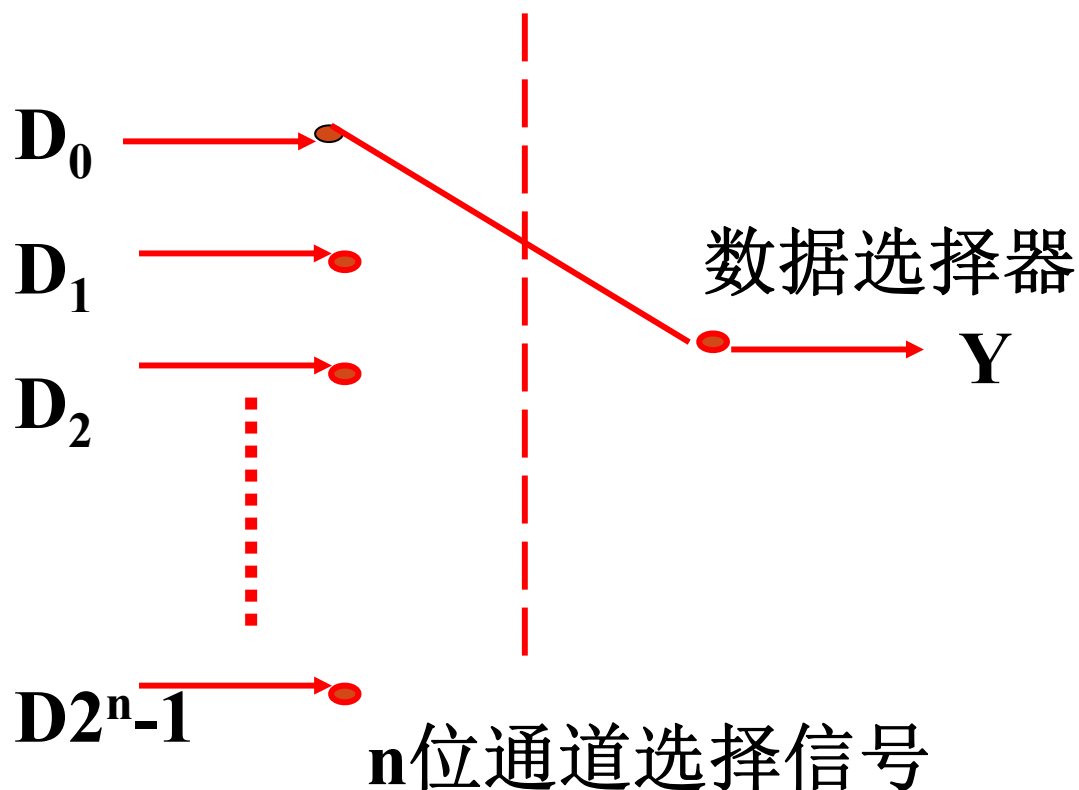


## § 4.4.3 数据选择器

**定义：** 根据需要将多路信号中选择一路送到公共数据线上的逻辑电路(又称**多路开关**).

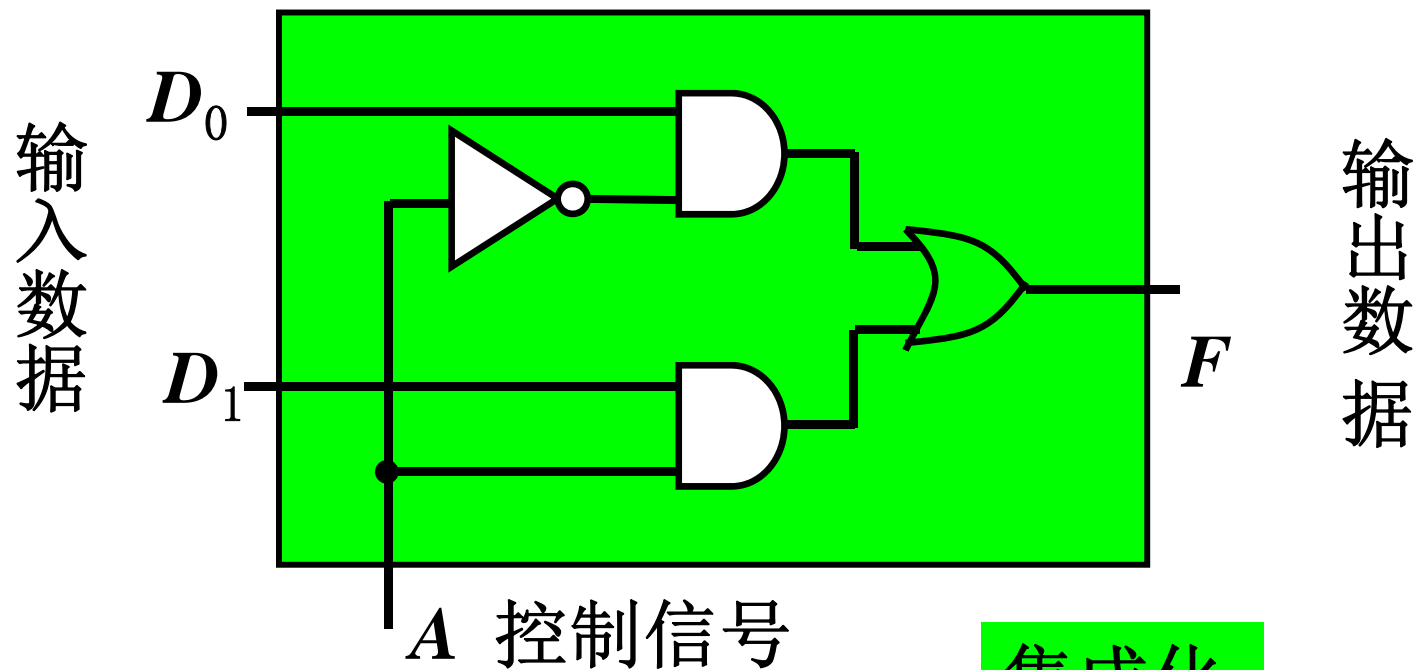
输入端：  $2^n$  个

输出端： 1 个





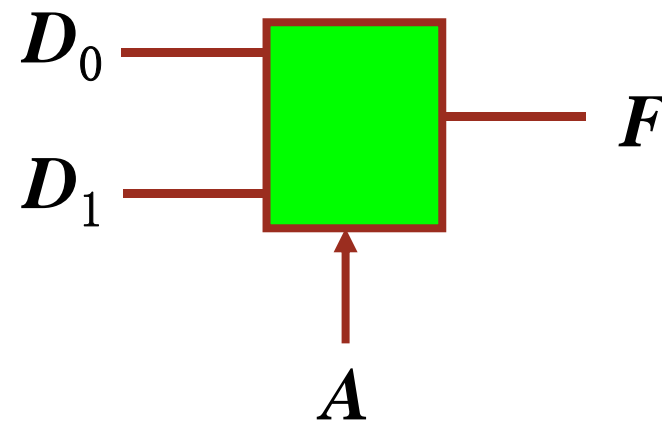
# 1、2选1数据选择器



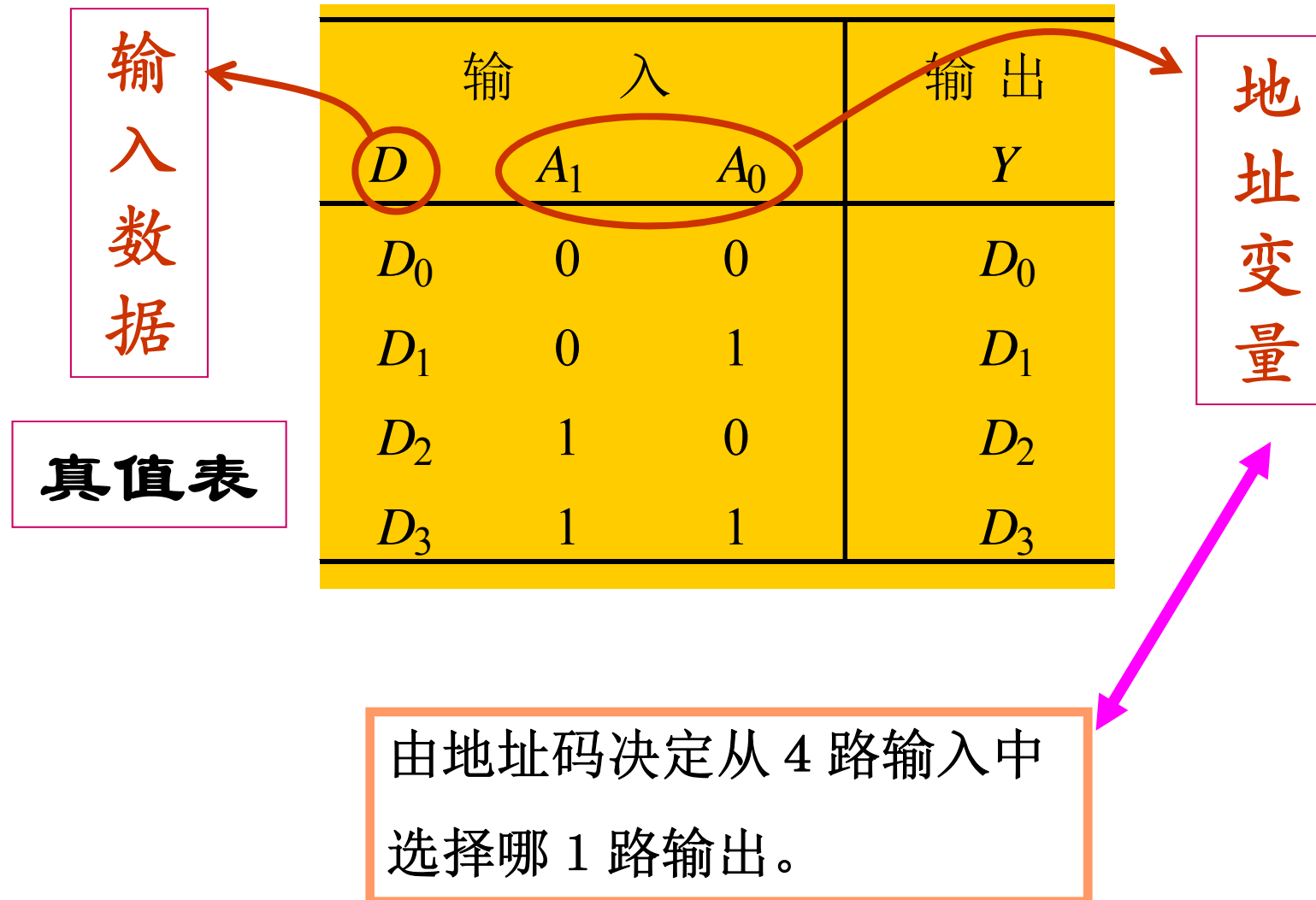
$$F = A'D_0 + AD_1$$

$A$	$F$
0	$D_0$
1	$D_1$

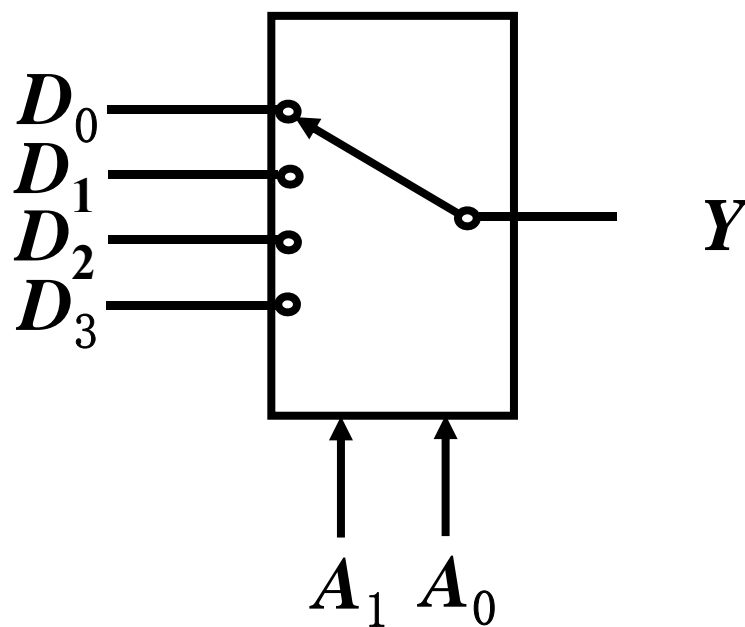
集成化



## 2、4选1数据选择器



即：



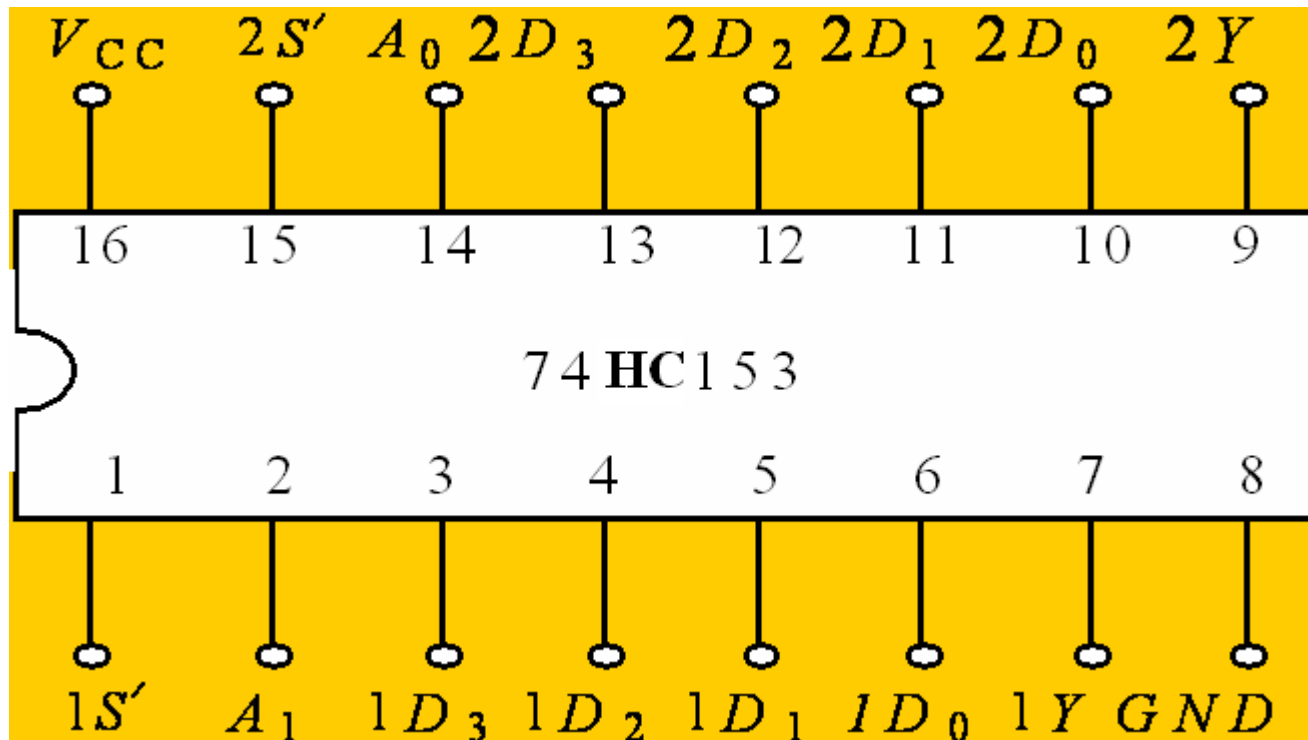
$A_1$	$A_0$	$Y$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

$$Y = A_1' A_0' D_0 + A_1' A_0 D_1 + A_1 A_0' D_2 + A_1 A_0 D_3$$

## 集成电路数据选择器

型号: 74HC153

双4选1数据选择器

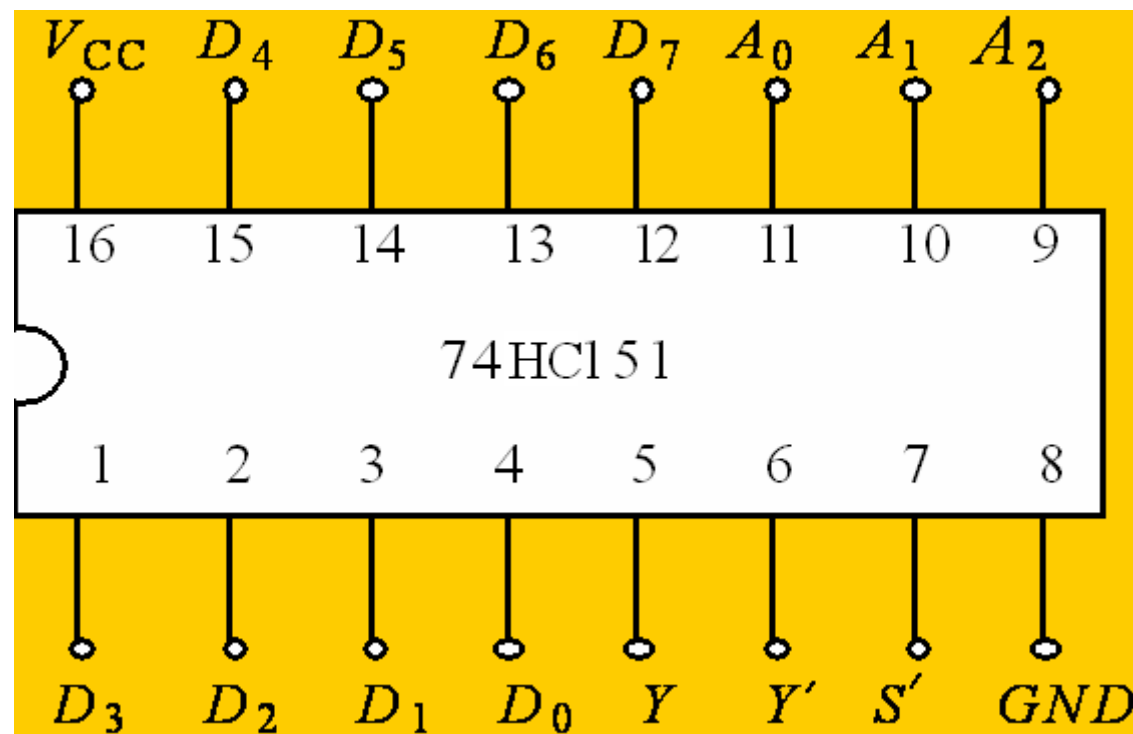


输入			输出
$A_1$	$A_0$	$S'_1$	$Y_1$
$\Phi$	$\Phi$	1	0
0	0	0	$D_{10}$
0	1	0	$D_{11}$
1	0	0	$D_{12}$
1	1	0	$D_{13}$

$$Y_1 = [(A'_1 A'_0) D_{10} + (A'_1 A_0) D_{11} + (A_1 A'_0) D_{12} + (A_1 A_0) D_{13}] \cdot S_1$$

$$Y_2 = [(A'_1 A'_0) D_{20} + (A'_1 A_0) D_{21} + (A_1 A'_0) D_{22} + (A_1 A_0) D_{23}] \cdot S_2$$

集成8选1  
数据选择  
器  
74HC151



$S' = 1$  时，选择器被禁止，无论地址码是什么， $Y$ 总是等于 0

$S' = 0$

$$Y = (A_2' A_1' A_0') D_0 + (A_2' A_1' A_0) D_1 + (A_2' A_1 A_0') D_2 + (A_2' A_1 A_0) D_3 + (A_2 A_1' A_0') D_4 + (A_2 A_1' A_0) D_5 + (A_2 A_1 A_0') D_6 + (A_2 A_1 A_0) D_7$$

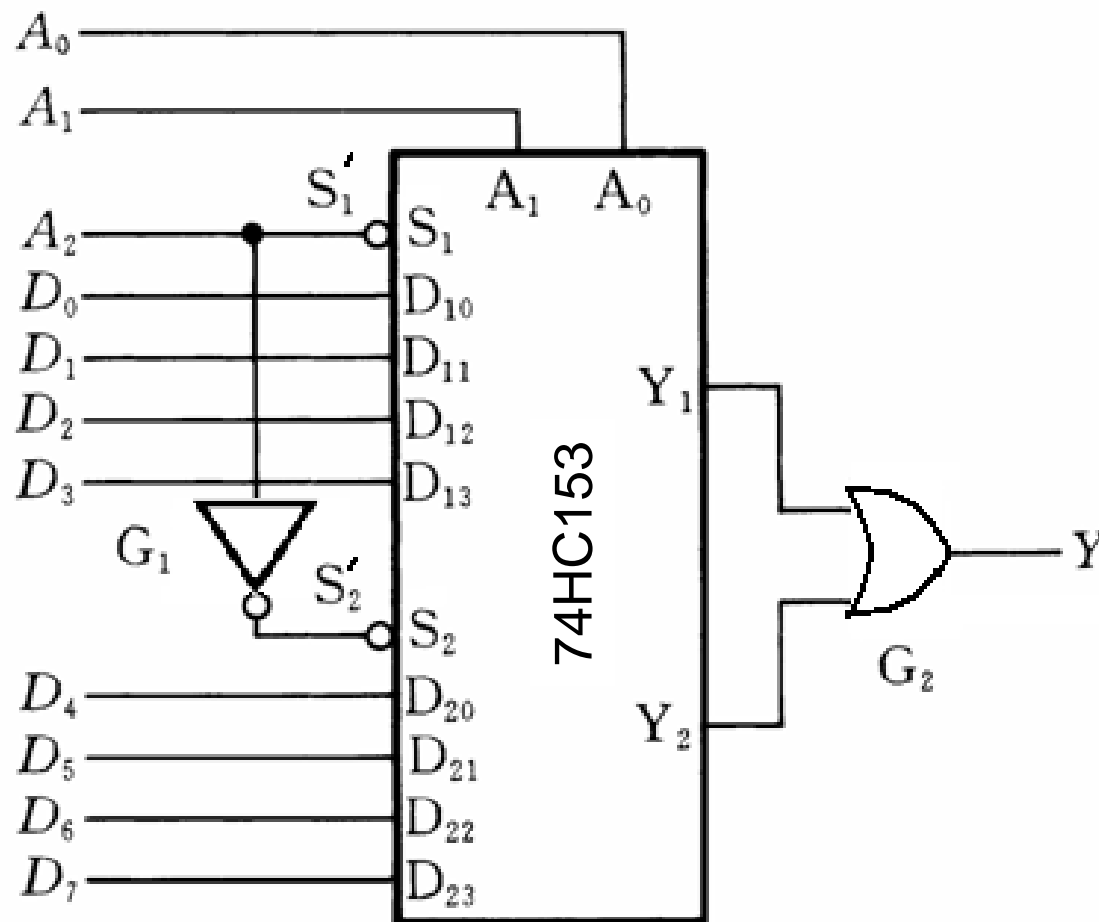
74HC151的真值表

输 入					输 出	
$D$	$A_2$	$A_1$	$A_0$	$S'$	$Y$	$Y'$
$\times$	$\times$	$\times$	$\times$	1	0	1
$D_0$	0	0	0	0	$D_0$	$D'_0$
$D_1$	0	0	1	0	$D_1$	$D'_1$
$D_2$	0	1	0	0	$D_2$	$D'_2$
$D_3$	0	1	1	0	$D_3$	$D'_3$
$D_4$	1	0	0	0	$D_4$	$D'_4$
$D_5$	1	0	1	0	$D_5$	$D'_5$
$D_6$	1	1	0	0	$D_6$	$D'_6$
$D_7$	1	1	1	0	$D_7$	$D'_7$

扩展:

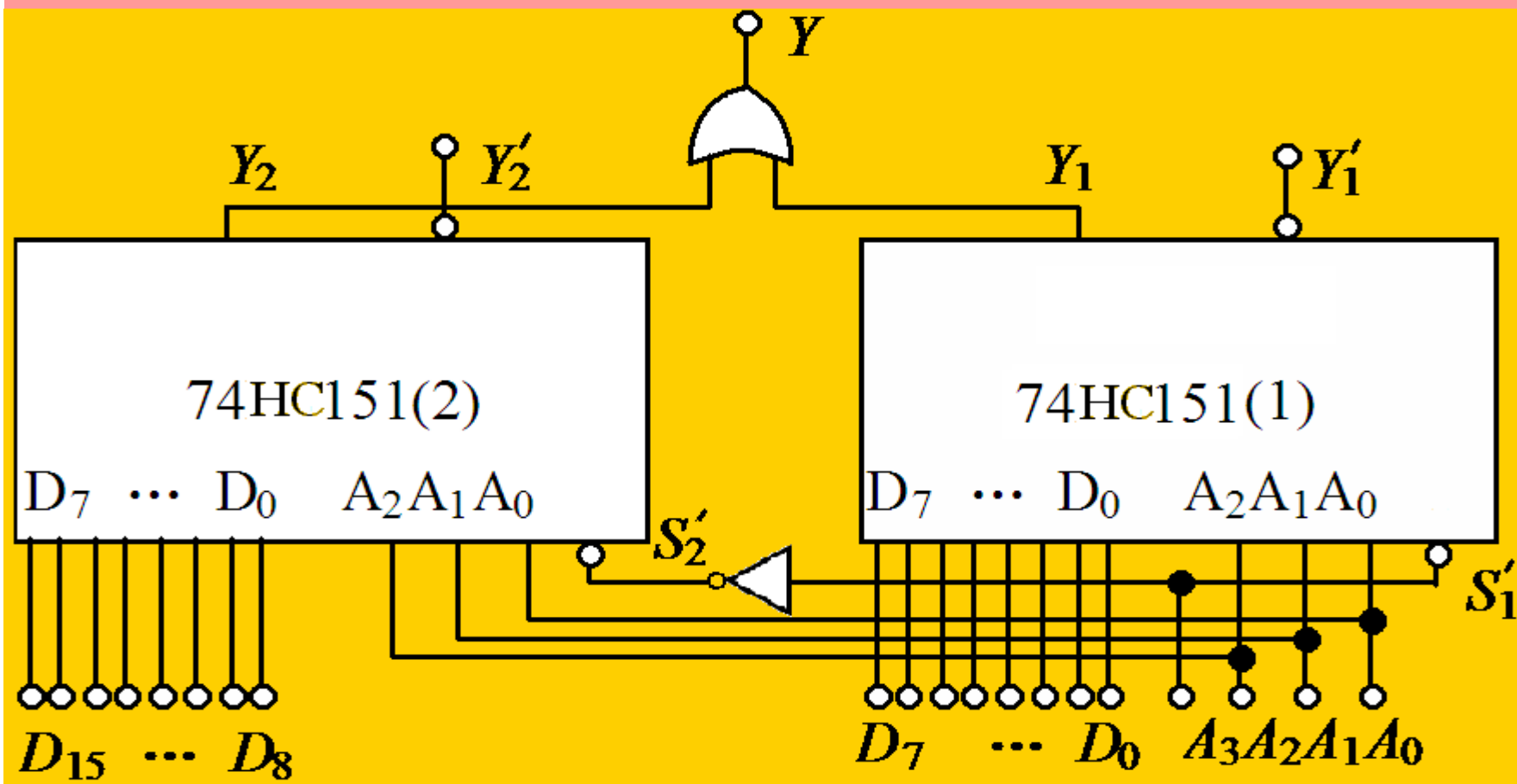
### 例4.3.4

用双4选1数据选择器构成8选1数据选择器。



$A_2=1$ 时，下边一半数据选择器工作，数据 $D_4 \sim D_7$ 选择一路输出。

## 2片8选1数据选择器74LS151构成16选1的数据选择器



$A_3 = 0$  时,  $S_1' = 0$ 、 $S_2' = 1$ , 片(2)禁止、片(1)工作  
 $A_3 = 1$  时,  $S_1' = 1$ 、 $S_2' = 0$ , 片(1)禁止、片(2)工作

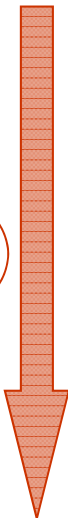


# 用数据选择器设计组合逻辑电路

## 步骤:

- 1.列出所求逻辑函数的真值表，写出其最小项表达式。
- 2.根据上述函数包含的变量数，选定数据选择器。
- 3.对照比较所求逻辑函数式和数据选择器的输出表达式确定选择器输入变量的表达式或取值。
- 4.按照求出的表达式或取值连接电路，画电路连线图。

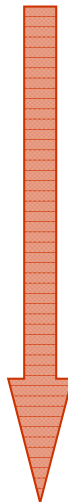
## 逻辑函数

1   $n$ 个地址变量的数据选择器，不需要增加门电路，最多可实现 $n+1$ 个变量的函数。

2  确定数据选择器

确定地址变量

$$L = A'B'C + A'BC' + AB$$

1  3个变量，选用4选1数据选择器。

2  选用74HC153

74HC153有两个地址变量。

$$A_1 = A, A_0 = B$$

3

## 公式法

函数的标准与或表达式:

$$\begin{aligned} L &= A'B'C + A'BC' + AB \\ &= A_1'A_0'C + A_1'A_0'C' + A_1A_0' \cdot 0 + A_1A_0 \cdot 1 \end{aligned}$$

4选1数据选择器输出信号的表达式:

$$Y = A_1'A_0'D_0 + A_1'A_0D_1 + A_1A_0'D_2 + A_1A_0D_3$$

比较L和Y, 得:

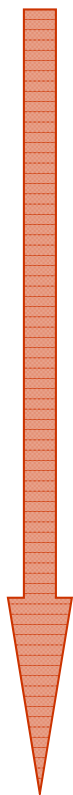
$$D_0 = C, D_1 = C', D_2 = 0, D_3 = 1$$

3

求  $D_i$

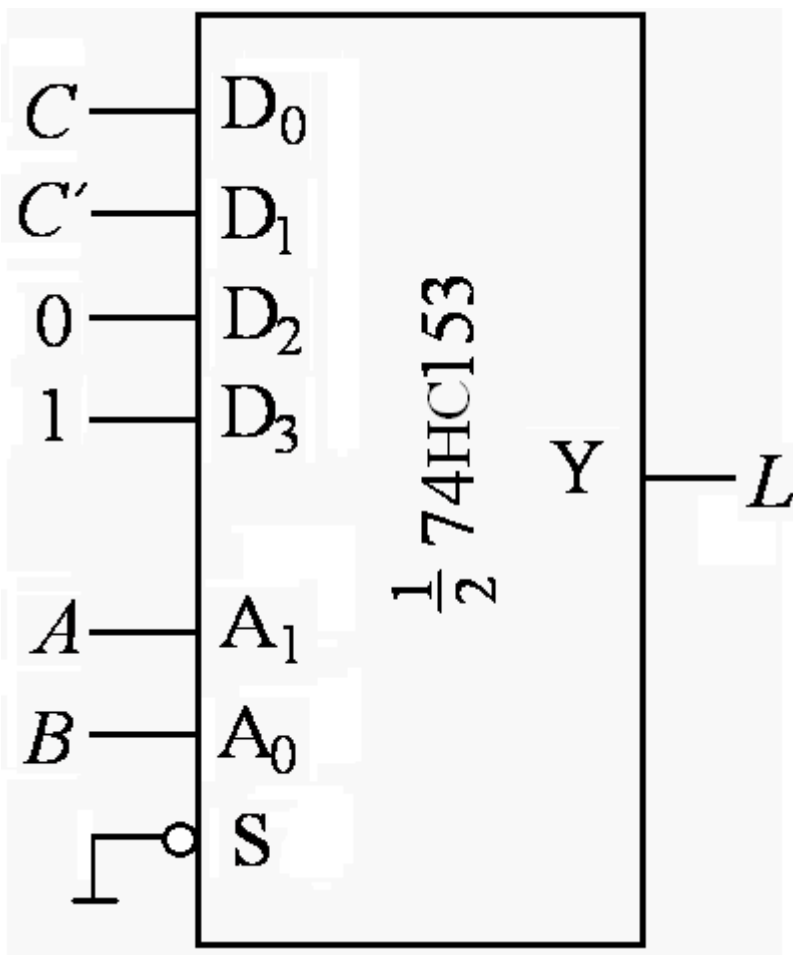
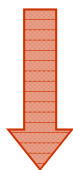


4



画连线图

4



### 例4.5.5

$$Z = A'B'C' + AC + A'BC$$

解：①写出最小项表达式

$$Z = A'B'C' + AC + A'BC = A'B'C' + AB'C + ABC + A'BC$$

②选用8选1数据选择器74HC151，当S'=0时，  
令A<sub>2</sub>=A、A<sub>1</sub>=B、A<sub>0</sub>=C,代入上式得：

$$Z = A'_2A'_1A'_0 + A_2A'_1A_0 + A_2A_1A_0 + A'_2A_1A_0$$

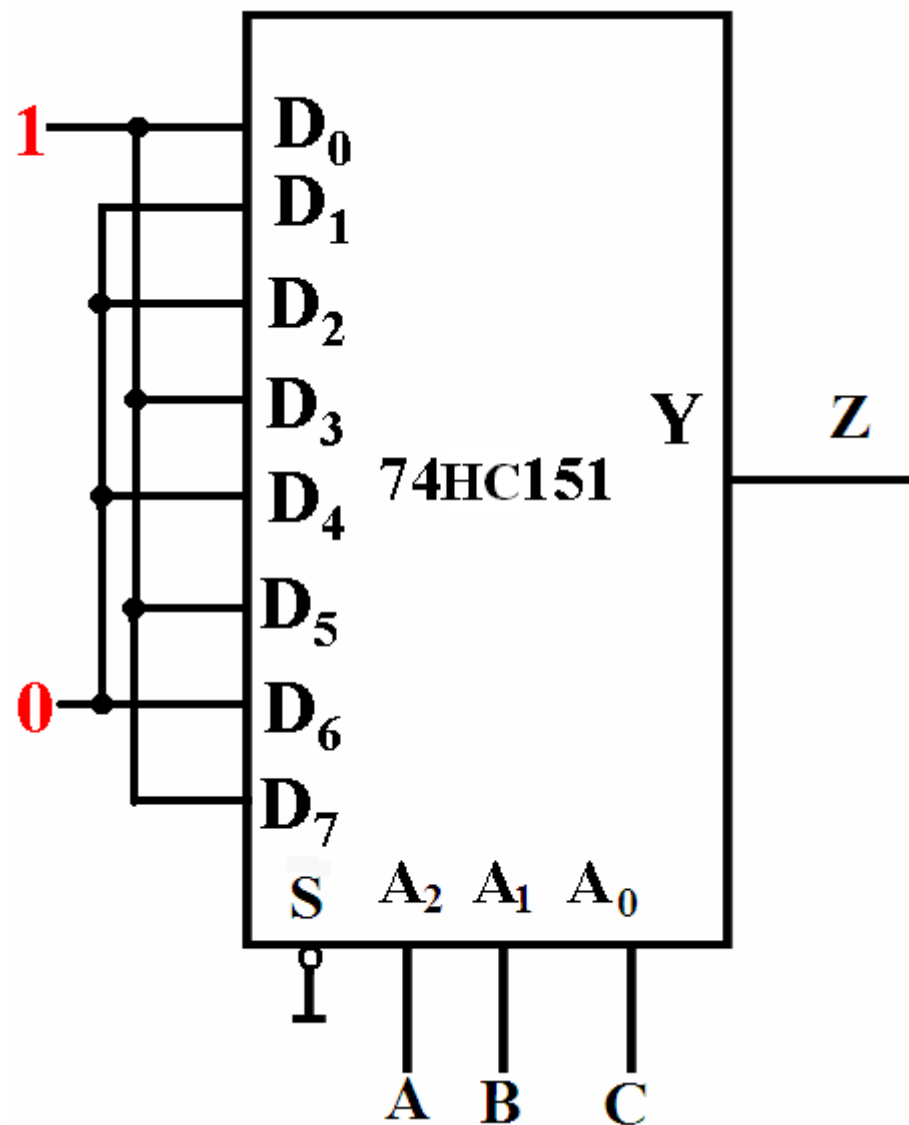
③对照74HC151输出表达式，求D<sub>i</sub>

$$Y = (A'_2A'_1A'_0)D_0 + (A'_2A'_1A_0)D_1 + (A'_2A_1A'_0)D_2 + (A'_2A_1A_0)D_3 \\ + (A_2A'_1A'_0)D_4 + (A_2A'_1A_0)D_5 + (A_2A_1A'_0)D_6 + (A_2A_1A_0)D_7$$

比较L和Y，得：

$D_0 = 1$ 、 $D_1 = 0$ 、  
 $D_2 = 0$ 、 $D_3 = 1$ 、  
 $D_4 = 0$ 、 $D_5 = 1$ 、  
 $D_6 = 0$ 、 $D_7 = 1$

④画连线图



另解： ①写出最小项表达式

$$Z = A'B'C' + AC + A'BC = A'B'C' + AB'C + ABC + A'BC$$

②选用双4选1数据选择器74HC153其中的一半，  
当 $S_1'=0$ 时，令 $A_1=A$ 、 $A_0=B$ ，代入上式得：

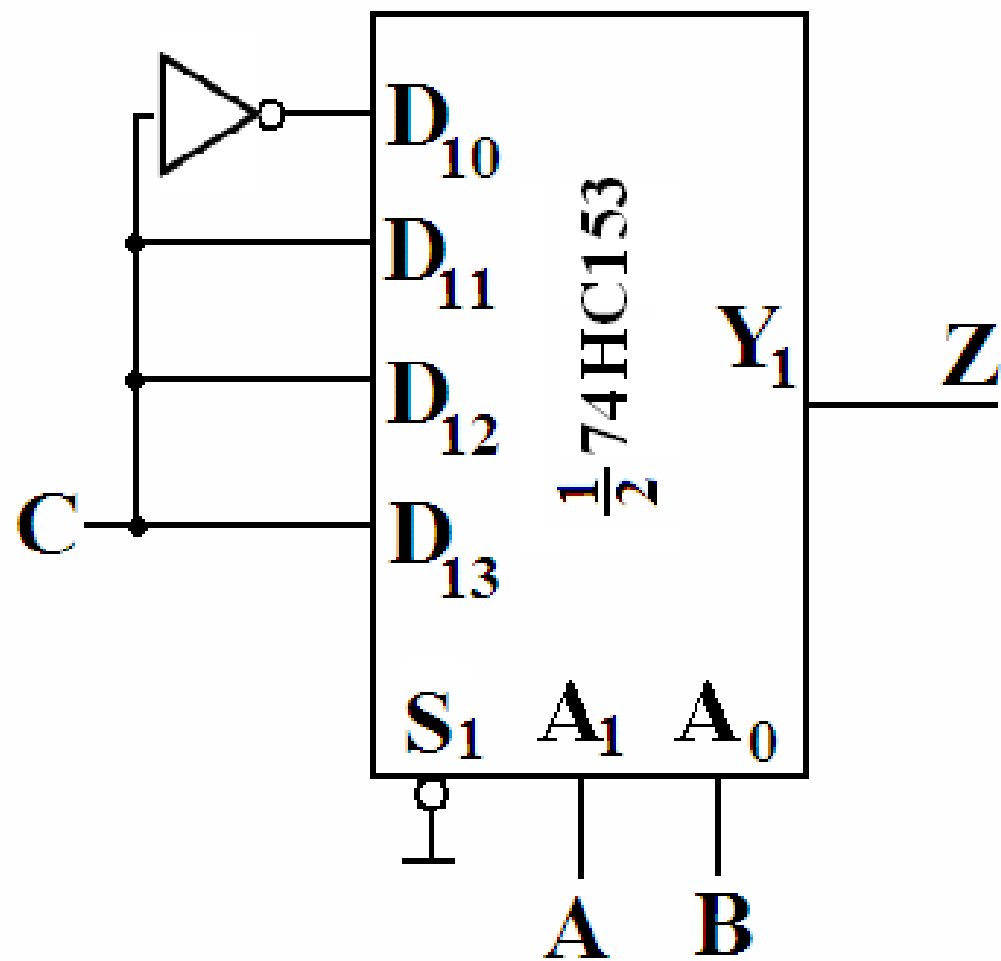
$$Z = A_1'A_0'C' + A_1A_0'C + A_1A_0C + A_1'A_0C$$

③对照74HC153输出表达式，求 $D_i$

$$Y_1 = [(A_1'A_0')D_{10} + (A_1'A_0)D_{11} + (A_1A_0')D_{12} + (A_1A_0)D_{13}] \cdot S_1$$

可得：  $D_{10}=C'$   $D_{11}=C$   $D_{12}=C$   $D_{13}=C$

④画连线图





### 例4.5.4（例4.3.2交通灯监视电 路）：列真值表

解：取红、黄、绿三盏灯分别用R、A、G表示，设灯亮为“1”，不亮为“0”；故障信号为输出变量用Z表示，规定正常为“0”，不正常为“1”。

#### ① 写逻辑函数式

$$Z = R'A'G' + R'AG + RA'G + RAG' + RAG$$

R	A	G	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

② 选2个地址输入端的4选1数据选择器(74HC153)

当 $S_1'=0$ 时, 令 $A_1=R$ 、 $A_0=A$ ,代入上式得:

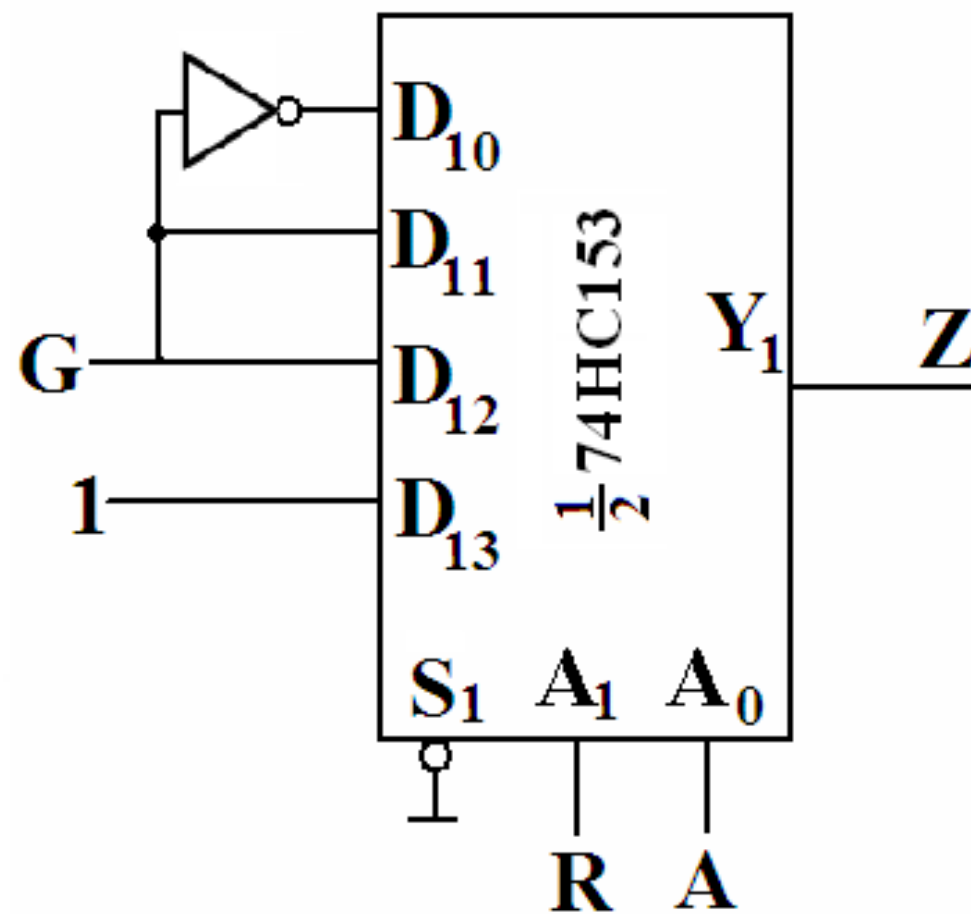
$$Z = A_1' A_0' G' + A_1' A_0 G + A_1 A_0' G + A_1 A_0 G'$$

③对照74HC153输出表达式, 求 $D_i$

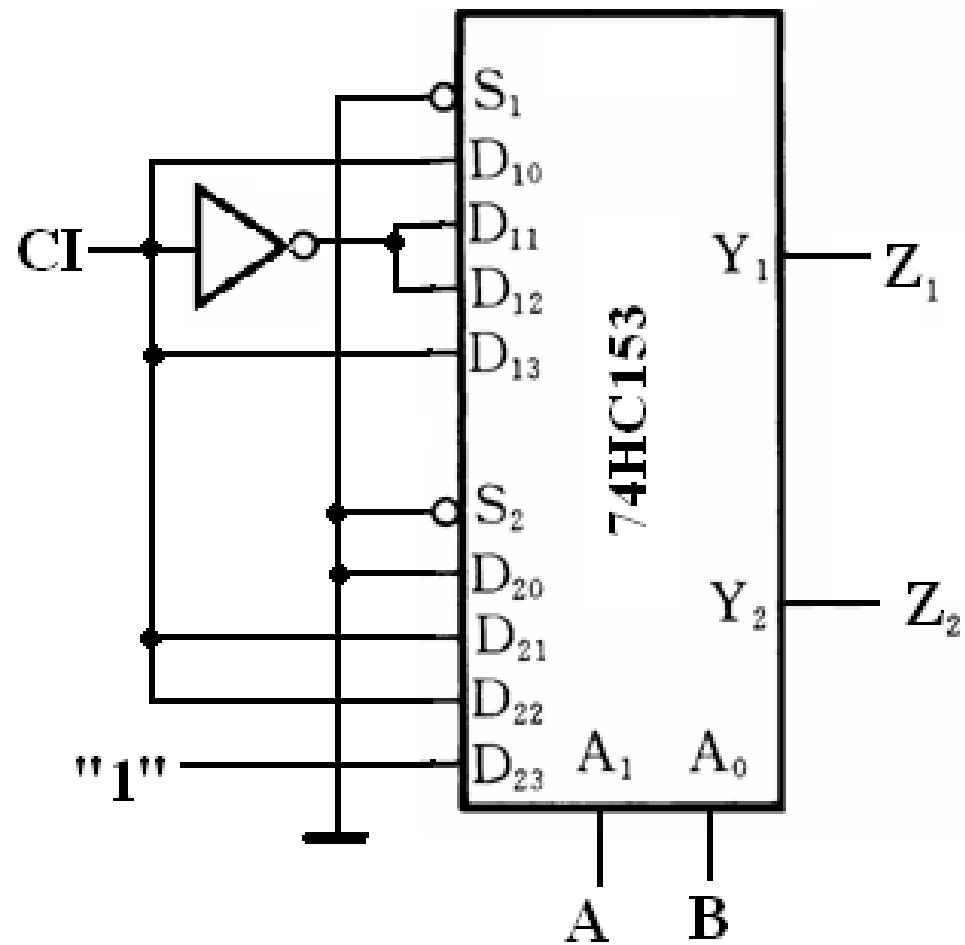
$$Y_1 = [(A_1' A_0') D_{10} + (A_1' A_0) D_{11} + (A_1 A_0') D_{12} + (A_1 A_0) D_{13}] \cdot S_1$$

可得:  $D_{10}=G' \quad D_{11}=G \quad D_{12}=G \quad D_{13}=1$

④画连线图



例：分析下图电路逻辑功能。

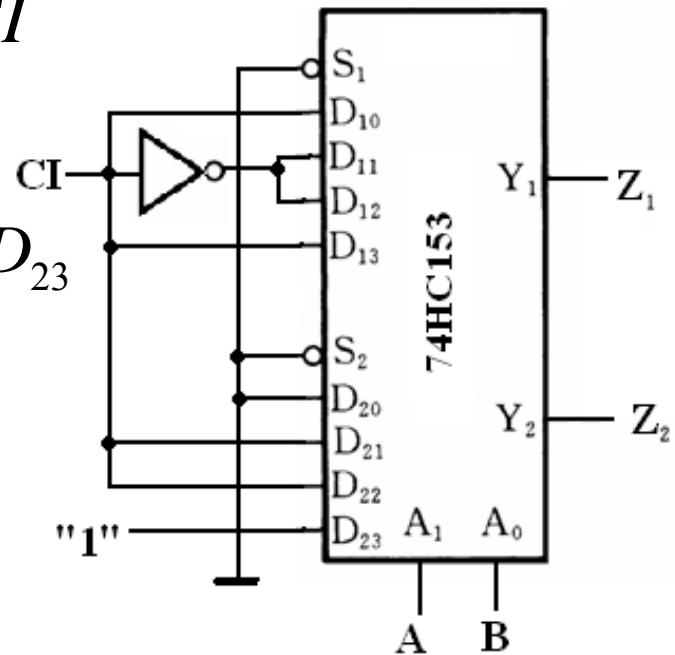


解：∵  $S_1' = S_2' = 0$

∴ 74HC153正常工作，且  $A_1 = A, A_0 = B$

$$\begin{aligned} Z_1 &= A_1' A_0' D_{10} + A_1' A_0 D_{11} + A_1 A_0' D_{12} + A_1 A_0 D_{13} \\ &= A' B' CI + A' B CI' + AB' CI' + ABCI \end{aligned}$$

$$\begin{aligned} Z_2 &= A_1' A_0' D_{20} + A_1' A_0 D_{21} + A_1 A_0' D_{22} + A_1 A_0 D_{23} \\ &= A' B' \cdot 0 + A' B CI + AB' CI + AB \cdot 1 \\ &= A' B CI + AB' CI + AB \end{aligned}$$



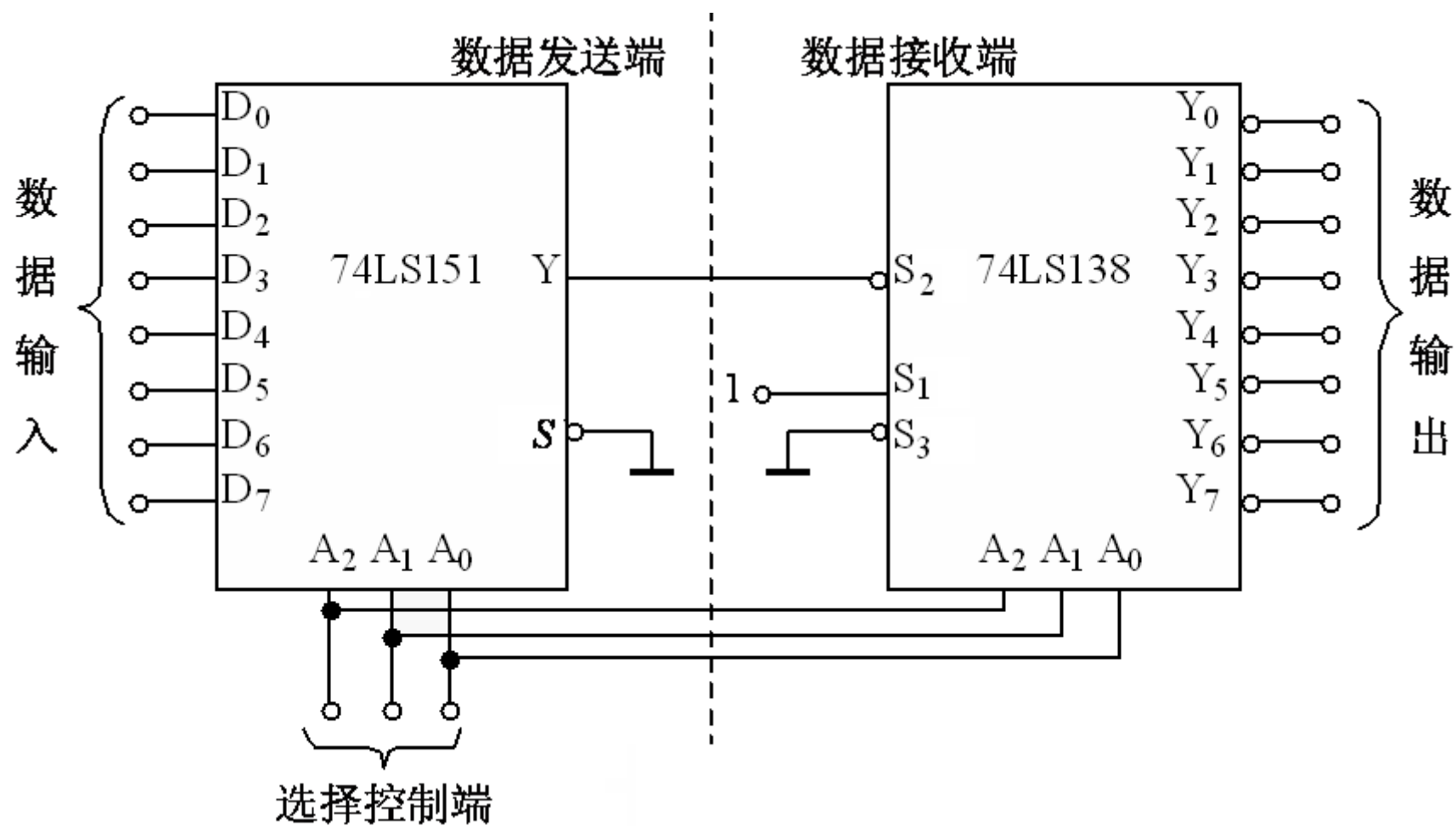
$A$	$B$	$CI$	$Z_1$	$Z_2$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Z_1 = A'B'CI + A'BCI' + AB'CI' + ABCI$$

$$Z_2 = A'BCI + AB'CI + AB$$

这是一个全  
加器电路

## 数据分配器和数据选择器一起构成数据分时传送系统



## § 4.4.4 加法器

举例：  $A=1101$ ,  $B=1001$ , 计算  $A+B$

$$\begin{array}{r} \phantom{+} 1\ 1\ 0\ 1 \\ + 1\ 0\ 0\ 1 \\ \hline 1\ 0\ 1\ 1\ 0 \end{array}$$



## 加法运算的基本规则：

- (1) 逢二进一。
- (2) 最低位是两个数最低位的相加，不需考虑进位。
- (3) 其余各位都是三个数相加，包括加数、被加数和低位来的进位。
- (4) 任何位相加都产生两个结果：本位和、向高位的进位。

## 1位加法器

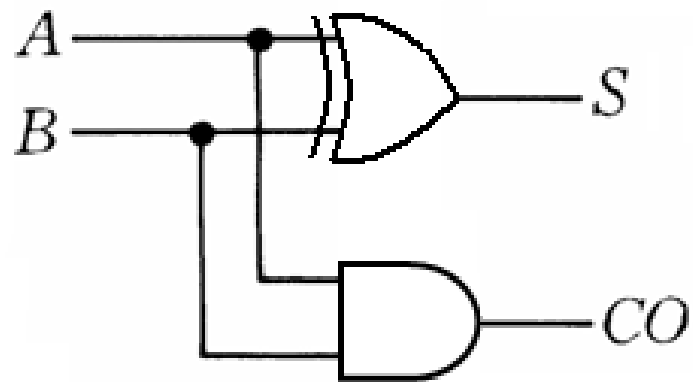
(1) 半加器: 半加运算不考虑从低位来的进位  
 $A$ ---加数;  $B$ ---被加数;  $S$ ---本位和;  $Co$ ---进位。

真值表

$A$	$B$	$Co$	$S$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

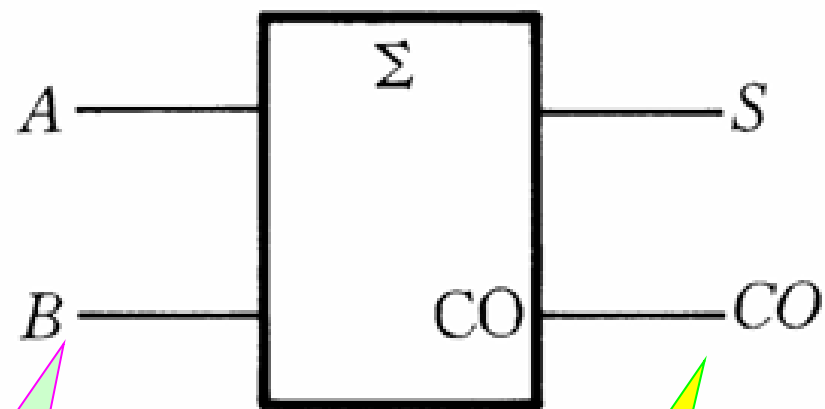
$$S = A'B + AB' = A \oplus B$$

$$Co = AB$$



(a)

逻辑图



(b)

逻辑符号

2个输入端

2个输出端

## (2) 全加器:

相加过程中, 既考虑加数、被加数又考虑低位的进位。

$A$ ---加数;  $B$ ---被加数;  $C_i$ ---低位的进位;  
 $S$ ---本位和;  $C_o$ ---进位。

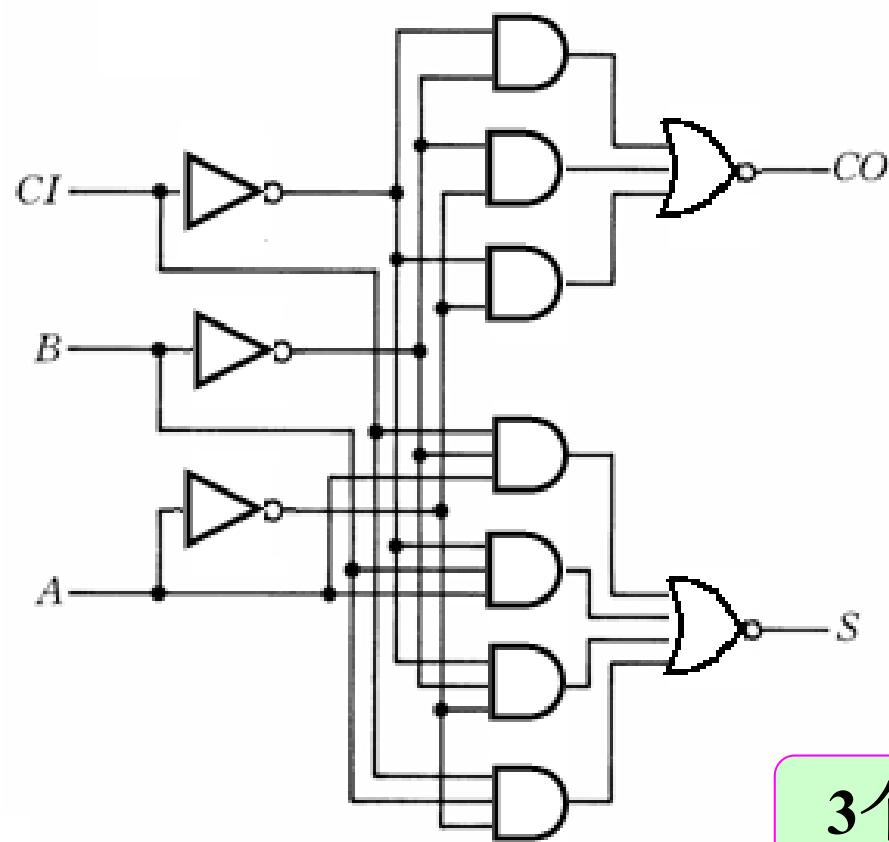
逻辑状态表见下页

$C_i$	$A$	$B$	$S$	$C_o$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = C_i' A' B + C_i' A B' + C_i A' B' + C_i A B = C_i \oplus A \oplus B$$

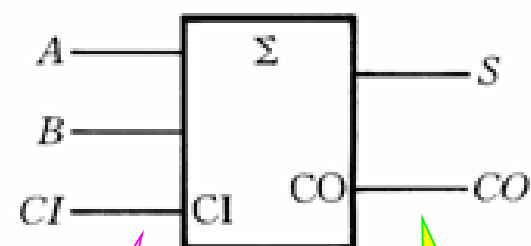
$$C_o = AB + C_i(A \oplus B) = AB + C_i A + C_i B$$

课本上采用了圈0的方法(p193)



(a)

双全加器74LS183



(b)

3个输入端

2个输出端

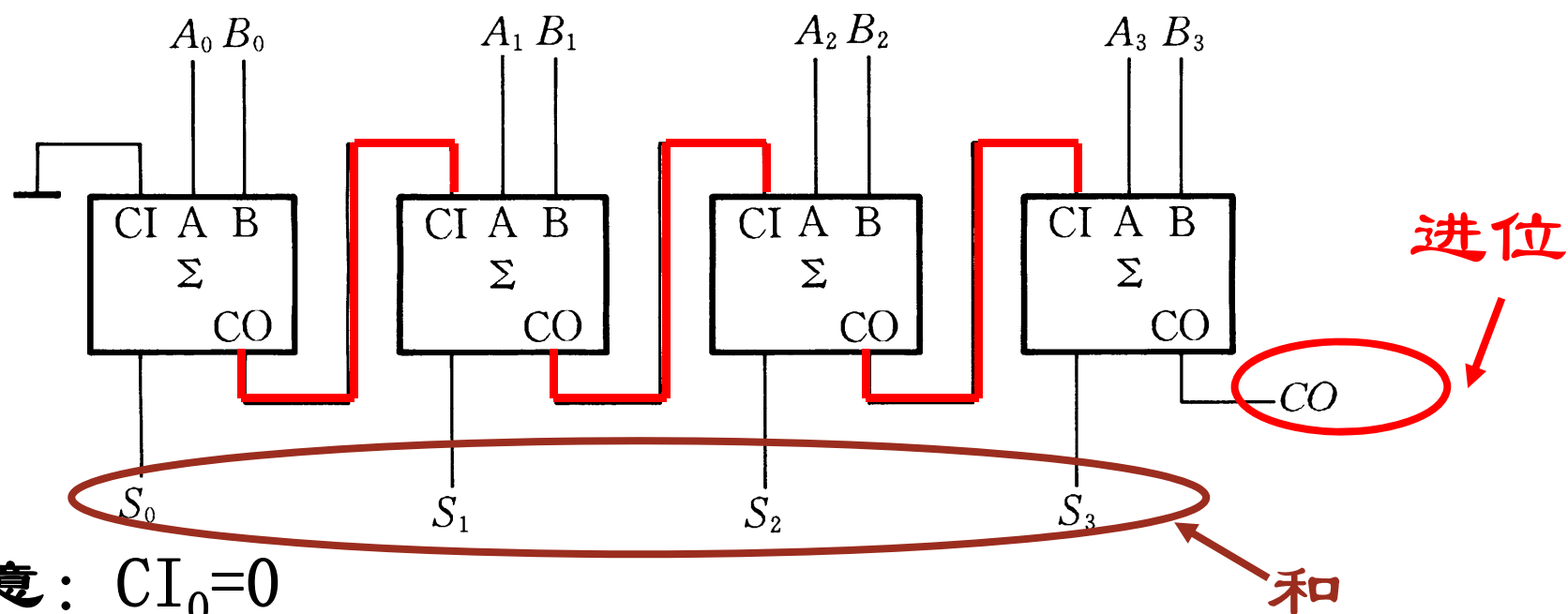
# 多位加法器

## (1) 串行进位加法器

低位全加器进位输出

↓  
高位全加器进位输入

如图：用全加器实现4位二进制数相加。



## (2) 超前进位加法器

**进位生成项**

$$G_i = A_i B_i$$

**进位传递条件**

$$P_i = A_i + B_i$$

**进位表达式**

$$(CO)_i = A_i B_i + (A_i + B_i)(CI)_i = G_i + P_i(CI)_i$$

**和表达式**

$$S_i = A_i \oplus B_i \oplus (CI)_i$$

$$\begin{cases} S_0 = A_0 \oplus B_0 \oplus (CI)_0 \\ (CO)_0 = G_0 + P_0(CI)_0 \end{cases}$$

$$\begin{cases} S_1 = A_1 \oplus B_1 \oplus (CI)_1 \\ (CO)_1 = G_1 + P_1(CI)_1 = G_1 + P_1 G_0 + P_1 P_0 (CI)_0 \end{cases}$$

$$\begin{cases} S_2 = A_2 \oplus B_2 \oplus (CI)_2 \\ (CO)_2 = G_2 + P_2(CI)_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 (CI)_0 \end{cases}$$

$$\begin{cases} S_3 = A_3 \oplus B_3 \oplus (CI)_3 \\ (CO)_3 = G_3 + P_3(CI)_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 (CI)_0 \end{cases}$$

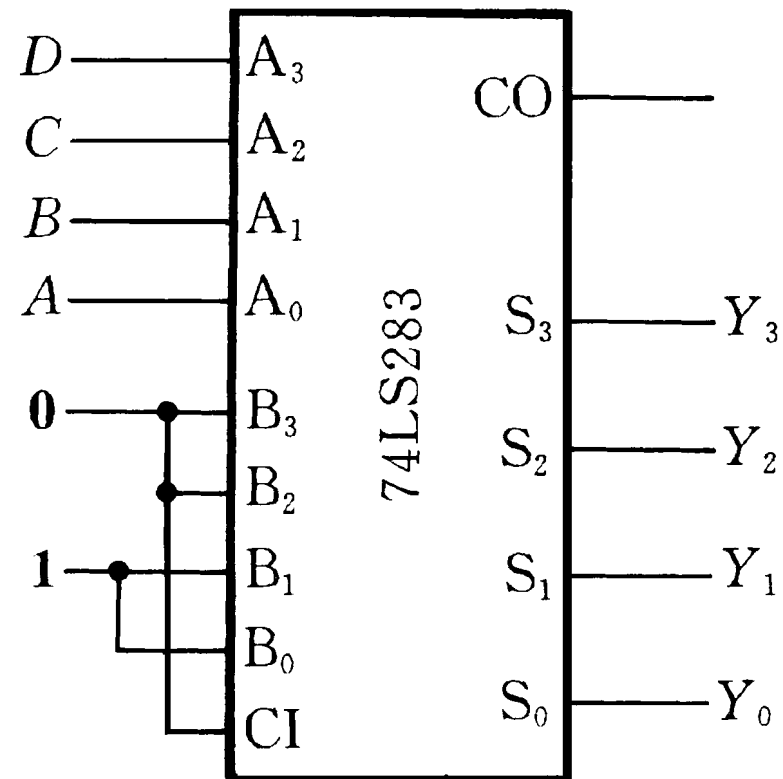


### 应用举例4.3.7

解： **BCD码+0011=余3码**

设输入8421码用变量DCBA表示，输出余三码用变量 $Y_3Y_2Y_1Y_0$ 表示。  
则有

$$Y_3Y_2Y_1Y_0 = DCBA + 0011$$



## § 4.4.5 数值比较器

**定义：**对两数**A**、**B**（可以是一位，也可能是多位）进行大小比较的逻辑电路。比较的结果有 **$A > B$** 、 **$A < B$** 、 **$A = B$** 三种结果。

## 一、1位数值比较器

设 $A > B$ 时 $Y_1 = 1$ ； $A < B$ 时 $Y_2 = 1$ ； $A = B$ 时 $Y_3 = 1$ 。

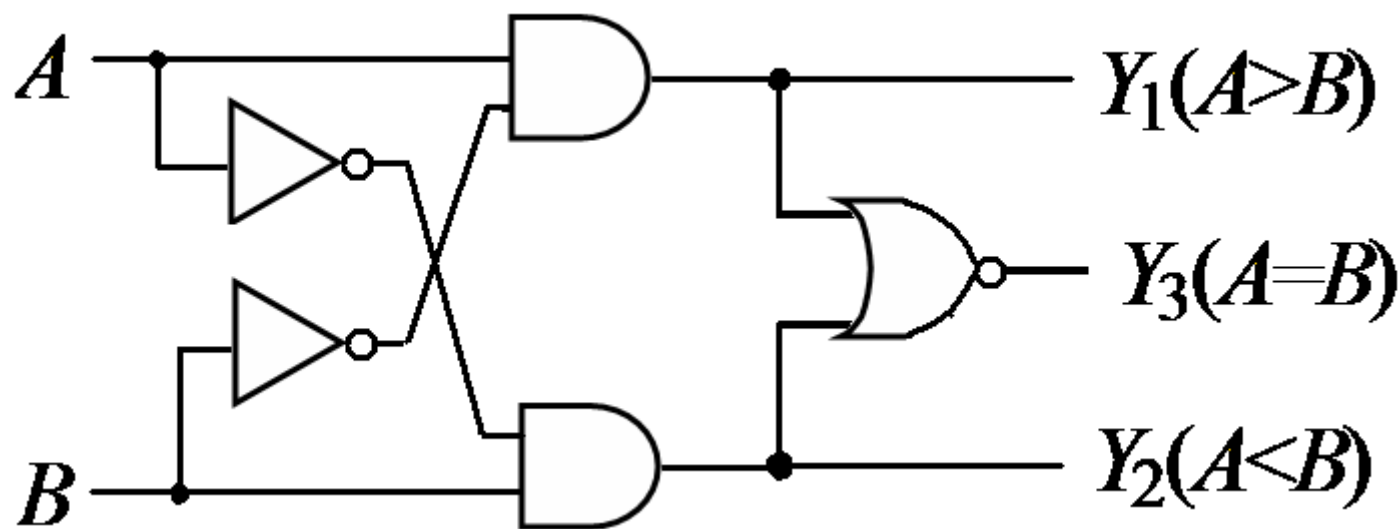
得1位数值比较器的真值表。

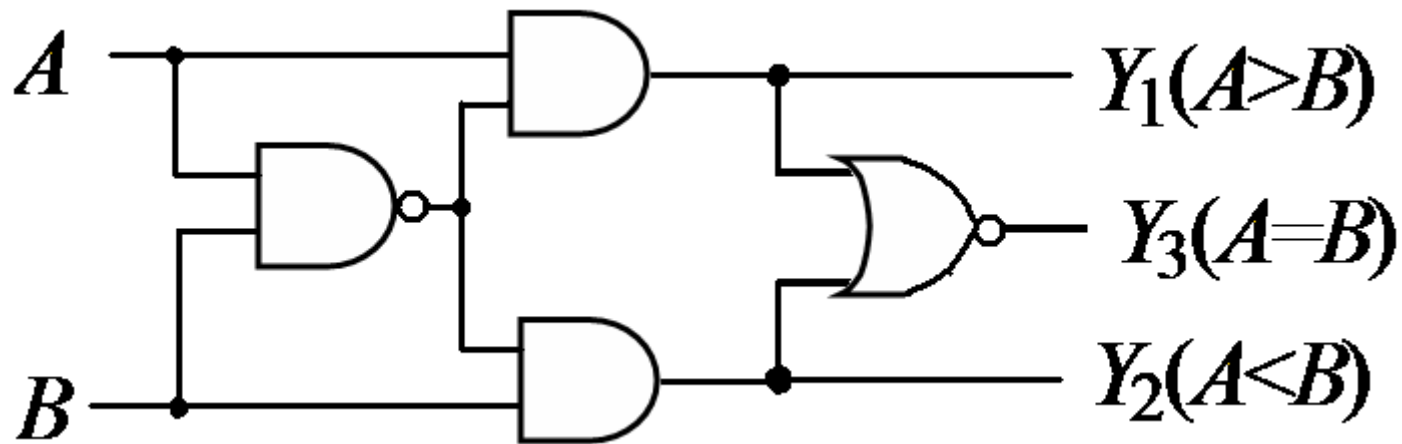
$A$	$B$	$Y_1(A > B)$	$Y_2(A < B)$	$Y_3(A = B)$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

逻辑表达式

$$\begin{cases} Y_1 = AB' \\ Y_2 = A'B \\ Y_3 = A'B' + AB = (A'B + AB')' \end{cases}$$

逻辑图





$$Y_1 = A \cdot (AB)' = A \cdot (A' + B') = AB'$$

$$Y_2 = B \cdot (AB)' = B \cdot (A' + B') = A'B$$

$$Y_3 = (AB' + A'B)' = AB + A'B'$$

## 二、多位数值比较器

比较原则：

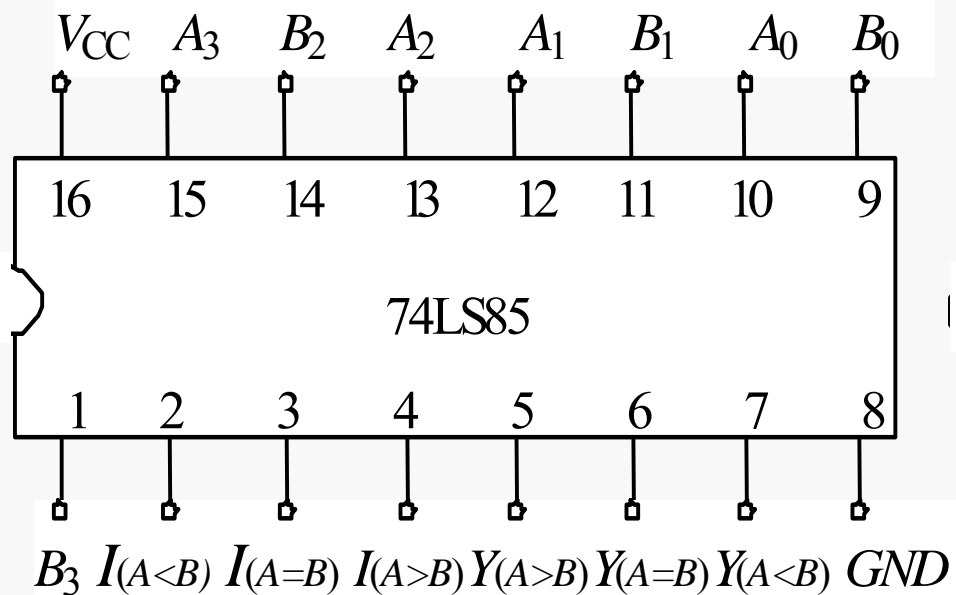
1. 先从高位比起,高位大的数值一定大。
2. 若高位相等，则再比较低位数，最终结果由低位的比较结果决定。

请根据这个原则设计一下：每位的比较应包括几个输入、输出？

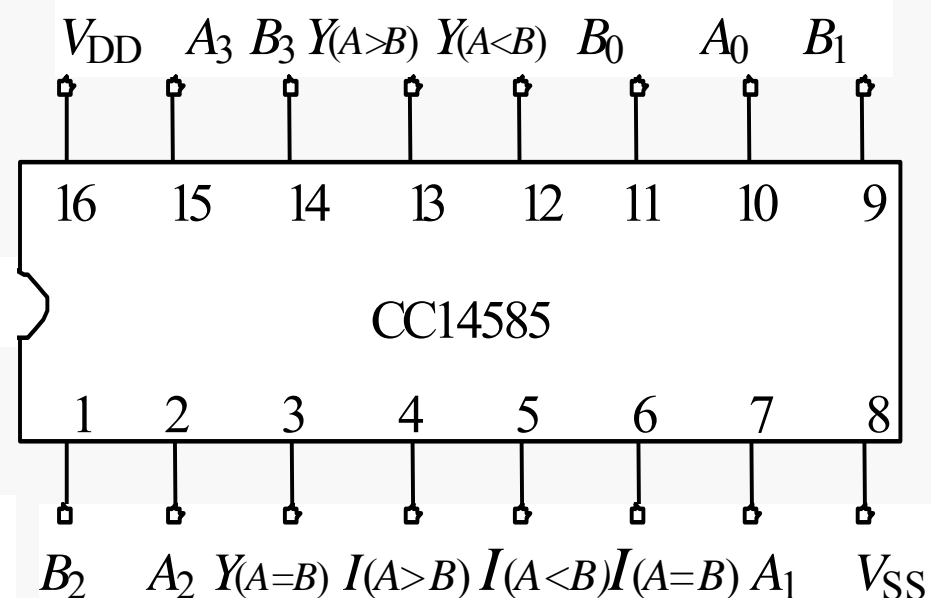
# 4位数值比较器

比 较 输 入				级 联 输 入			输 出		
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$I(A>B)$	$I(A<B)$	$I(A=B)$	$Y(A>B)$	$Y(A<B)$	$Y(A=B)$
$A_3>B_3$	×	×	×	×	×	×	1	0	0
$A_3<B_3$	×	×	×	×	×	×	0	1	0
$A_3=B_3$	$A_2>B_2$	×	×	×	×	×	1	0	0
$A_3=B_3$	$A_2<B_2$	×	×	×	×	×	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	×	×	×	×	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	×	×	×	×	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	×	×	×	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	×	×	×	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	0	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	1	0	0	1

## 集成数值比较器



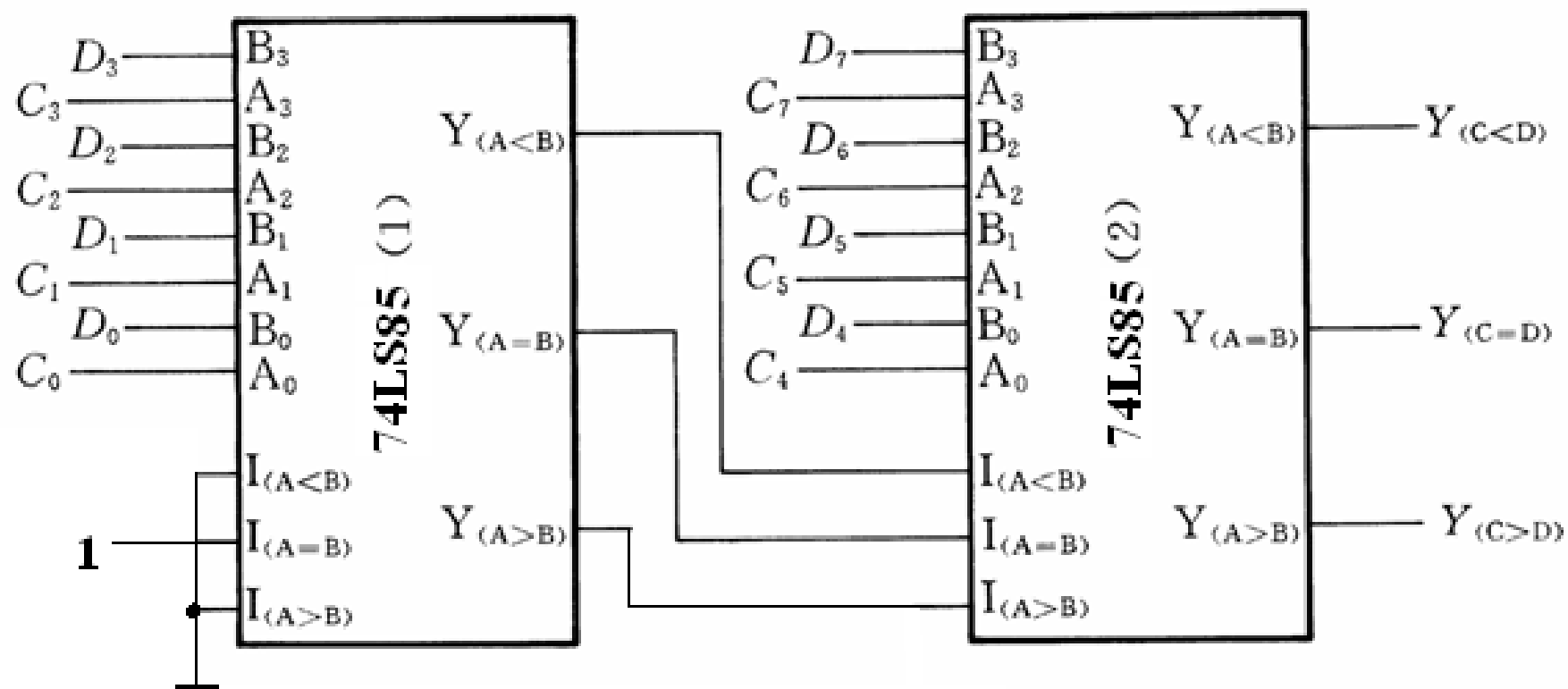
(a) TTL 数值比较器引脚图



(b) CMOS 数值比较器引脚图



**例4.5.3** 试用两片74LS85组成一个8位数值比较器。



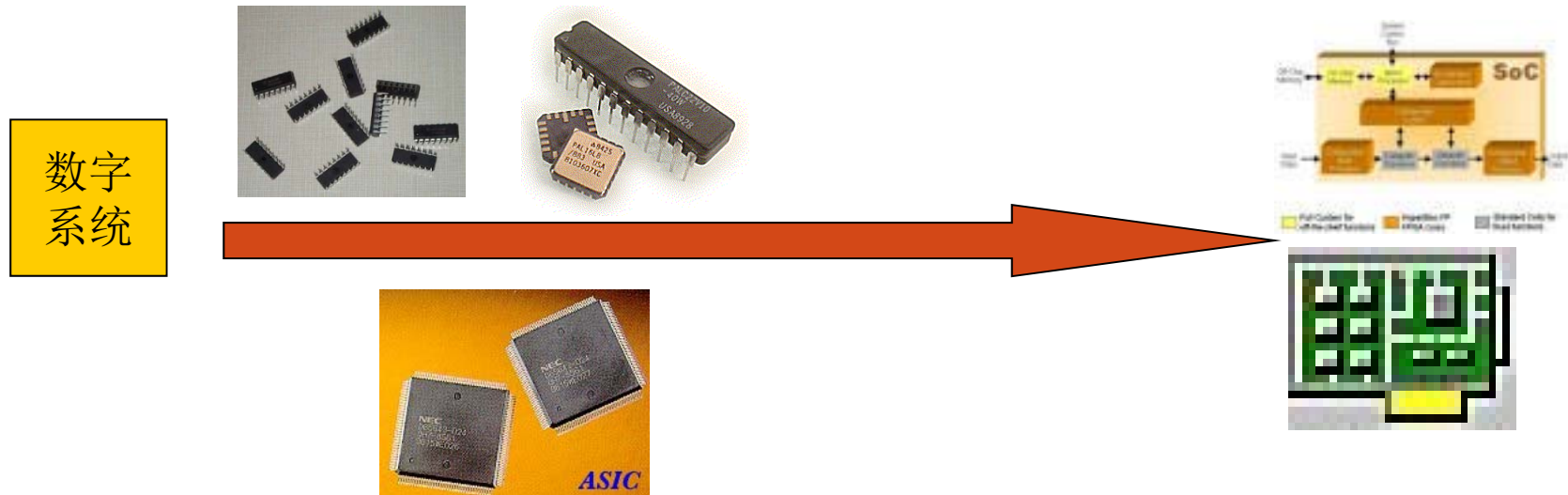
## 4.5 层次化和模块化的设计方法

---

## 4.6 可编程逻辑器件 (PLD, Programmable Logic Device)

### 一、PLD的基本特点

1. 数字集成电路从功能上分为通用型、专用型两大类



2. PLD的特点：是一种按通用器件来生产，但逻辑功能是由用户通过对器件编程来设定的

## 二、PLD的发展和分类

PROM是最早的PLD

1. FPLA 现场可编程逻辑阵列
2. PAL 可编程阵列逻辑
3. GAL 通用阵列逻辑
4. EPLD 可擦除的可编程逻辑器件
5. FPGA 现场可编程门阵列
6. CPLD 复杂可编程逻辑器件
7. ISP-PLD 在系统可编程的PLD

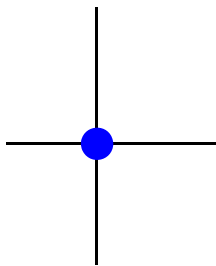
## 连接方式

● 可编程“接通”单元：  
它依靠用户编程来实现  
“接通”连接。

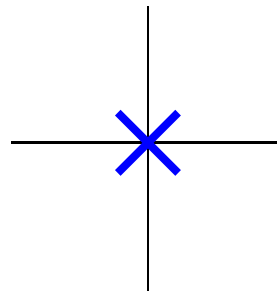
● 可编程“断开”单元：

编程实现断开状态。这种  
单元又称为被编程擦除单  
元。

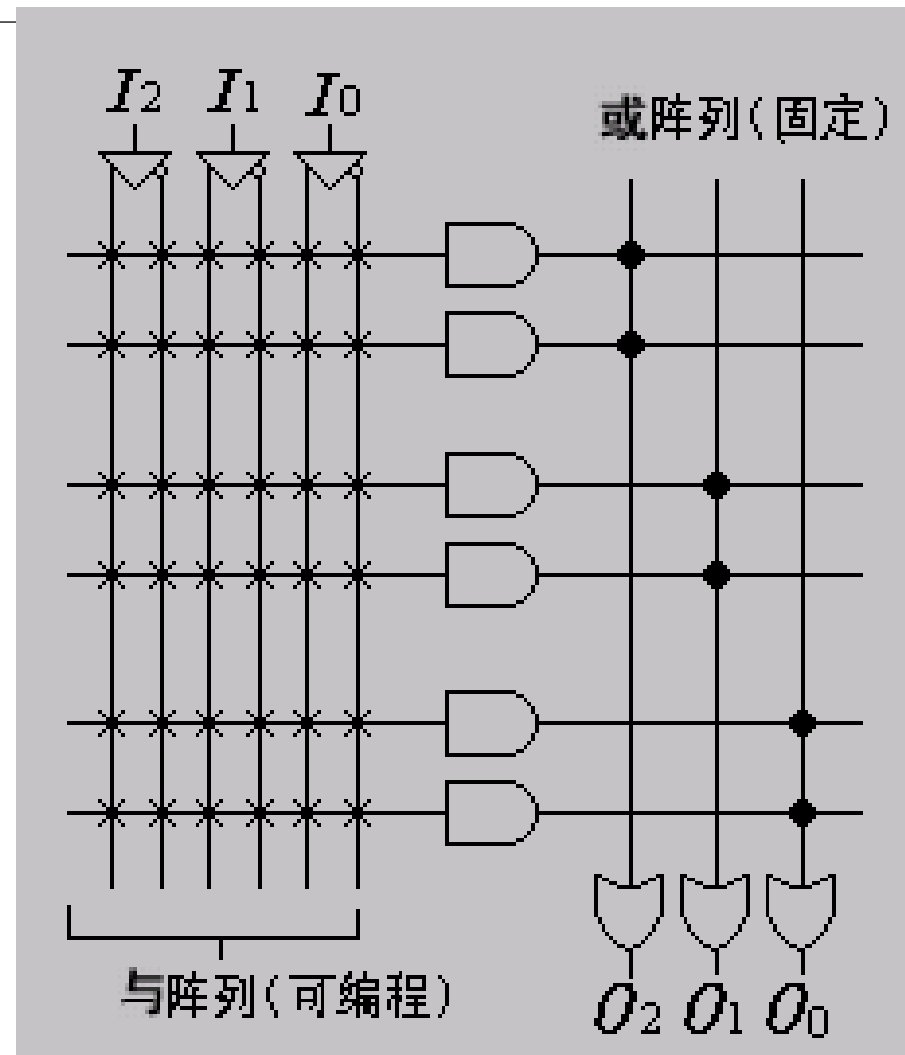
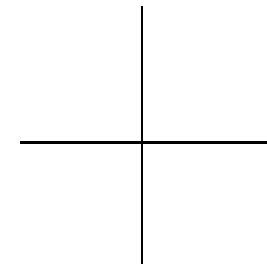
固定  
连接：



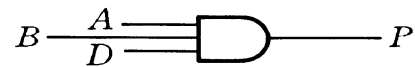
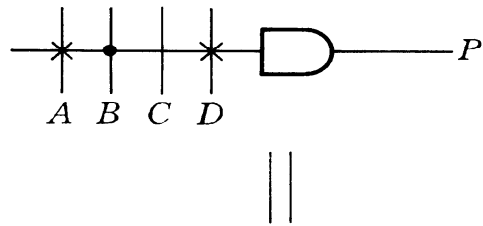
可  
编程  
连接：



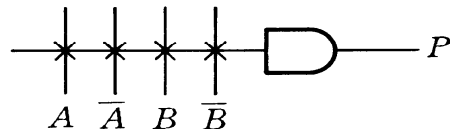
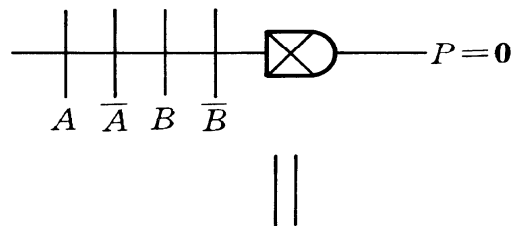
不  
连  
接：



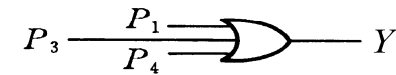
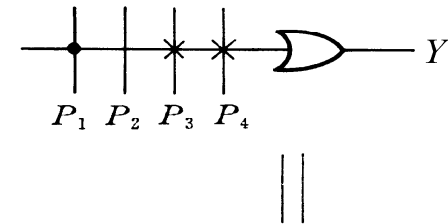
### 三、LSI中用的逻辑图符号



$$P = ABD$$



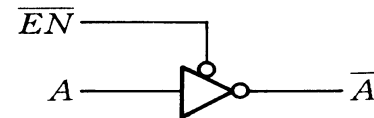
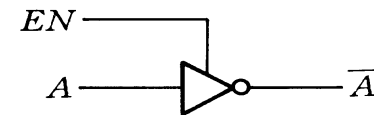
$$P = 0$$



$$Y = P_1 + P_3 + P_4$$



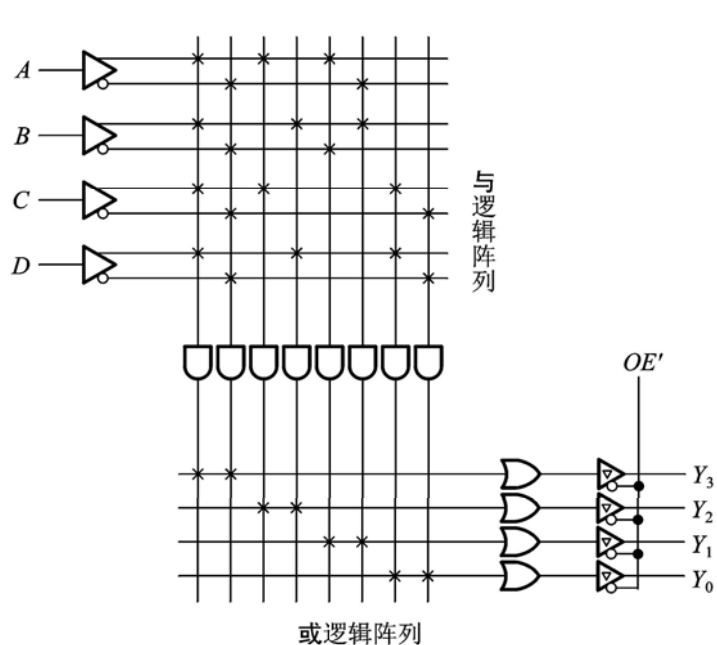
互补输出的缓冲器



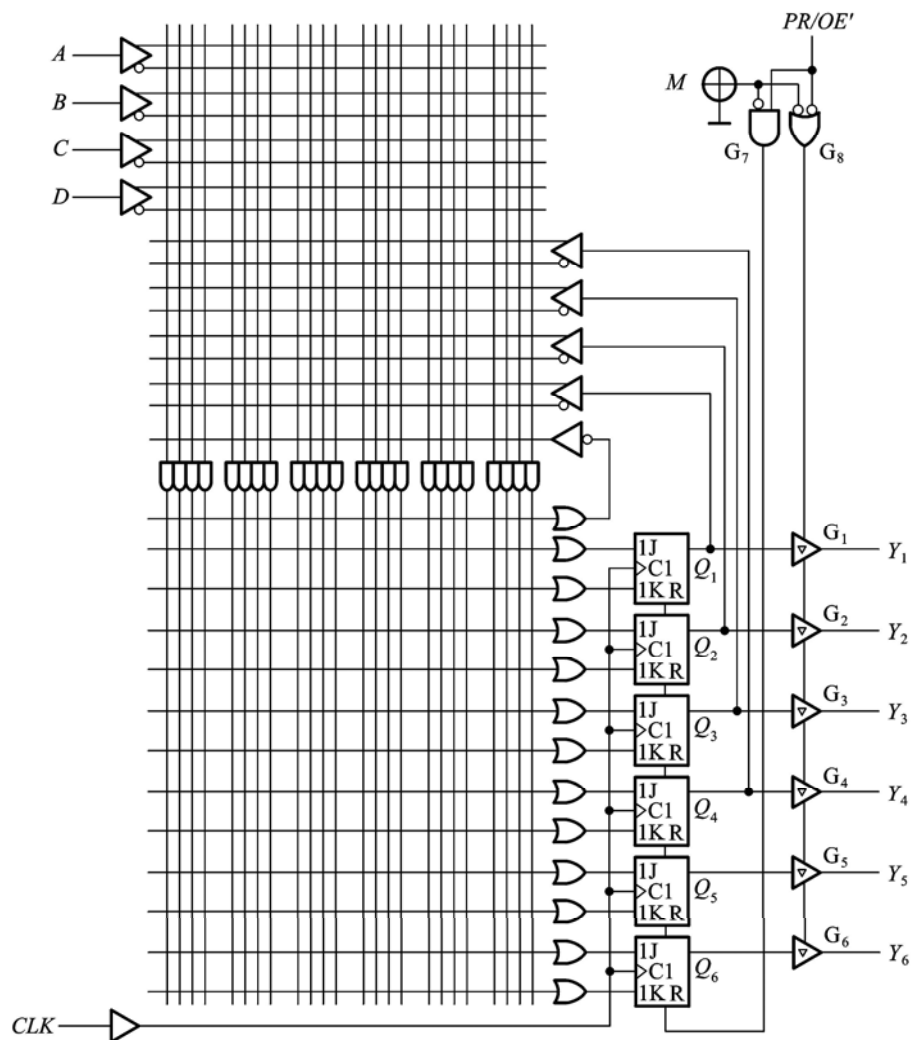
三态输出的缓冲器

# FPLA

## 组合电路和时序电路结构的通用形式



可编程的“与”阵列  
+ 可编程的“或”阵列



## 4.9 组合逻辑电路中的竞争—冒险现象

---

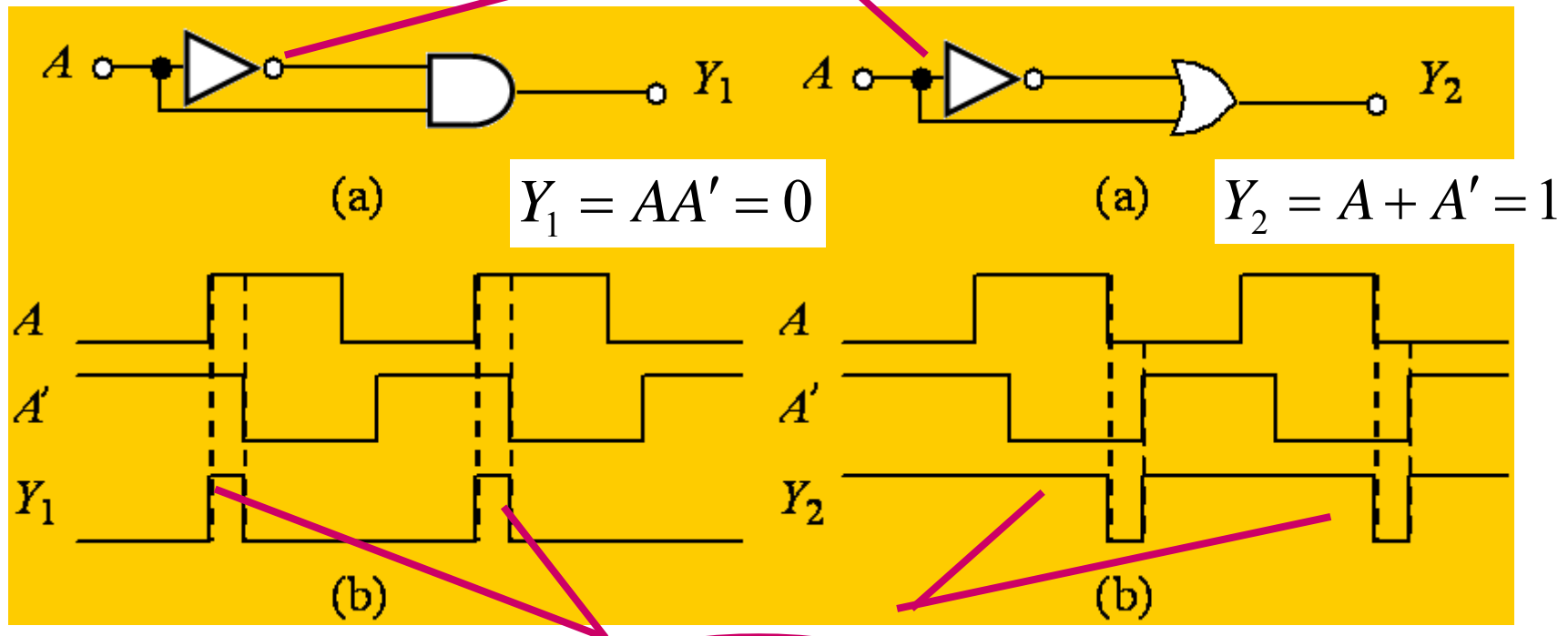
### 一.竞争—冒险的概念

在组合电路中，当输入信号的状态改变时，输出端可能会出现不正常的干扰信号，使电路产生错误的输出，这种现象称为 **竞争冒险**。



## 二.产生竞争—冒险的原因

原因：主要是门电路的延迟时间产生的。



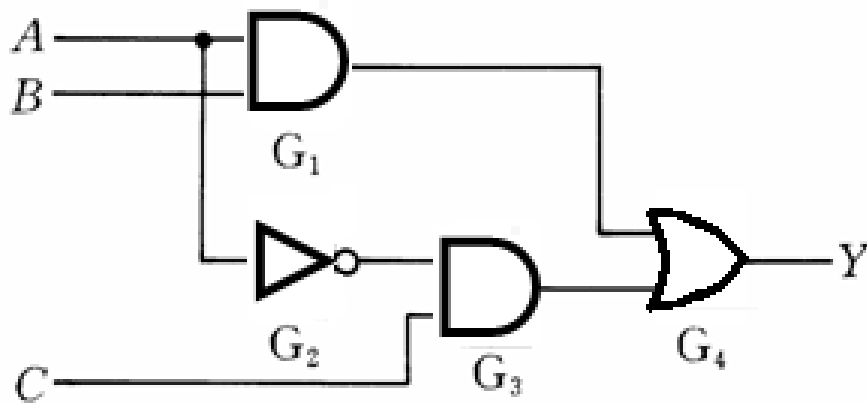
干扰信号

### 三. 检查竞争—冒险的方法

只要输出端的逻辑函数在一定条件下能简化成

$$Y = A + A' \quad \text{或} \quad Y = A \cdot A'$$

则可出现竞争—冒险现象。



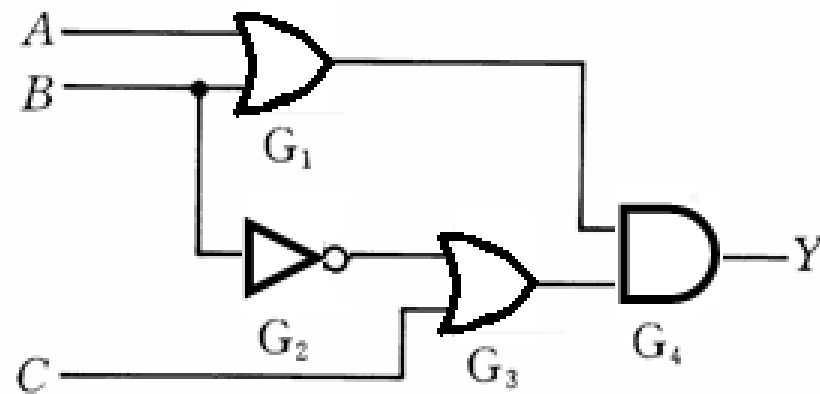
(a)

图(a)  $Y = AB + A'C$

当  $B=C=1$  时,

$$Y = A + A'$$

存在竞争冒险



(b)

图(b)  $Y = (A + B)(B' + C)$

当  $A=C=0$  时

$$Y = B \cdot B'$$

存在竞争冒险

## 四. 消除竞争—冒险的方法

1. 接入滤波电容
2. 引入选通脉冲
3. 修改逻辑设计（增加冗余项）

## 本章小结

①组合电路的特点：在任何时刻的输出只取决于当时的输入信号，而与电路原来所处的状态无关。实现组合电路的基础是逻辑代数和门电路。

②组合电路的逻辑功能可用逻辑图、真值表、逻辑表达式、卡诺图和波形图等5种方法来描述，它们在本质上是相通的，可以互相转换。

③组合电路的分析步骤：逻辑图→写出逻辑表达式→逻辑表达式化简→列出真值表→逻辑功能描述。

④组合电路的设计步骤：列出真值表→写出逻辑表达式或画出卡诺图→逻辑表达式化简和变换→画出逻辑图。

在许多情况下，如果用中、大规模集成电路来实现组合函数，可以取得事半功倍的效果。

# 组合逻辑电路的设计方法

## 电路功能描述

穷举法

1

## 真值表

**例：**设计一个楼上、楼下开关的控制逻辑电路来控制楼梯上的路灯，使之在上楼前，用楼下开关打开电灯，上楼后，用楼上开关关灭电灯；或者在下楼前，用楼上开关打开电灯，下楼后，用楼下开关关灭电灯。

1

设楼上开关为A，楼下开关为B，灯泡为Y。并设A、B闭合时为1，断开时为0；灯亮时Y为1，灯灭时Y为0。根据逻辑要求列出真值表。

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2

逻辑表达式  
或卡诺图

化简

3

最简与或  
表达式

4

逻辑变换

5

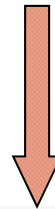
逻辑电路图

2

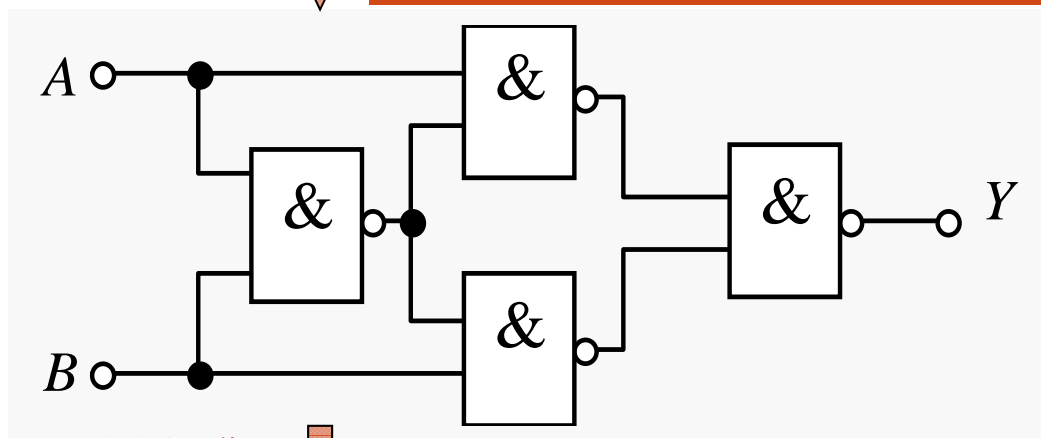
$$Y = A'B + AB'$$

已为最简与  
或表达式

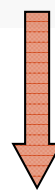
用与非  
门实现



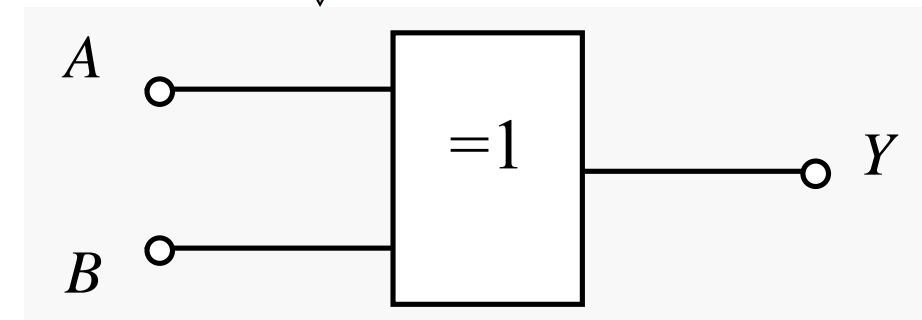
$$Y = ((A'B)' \cdot (AB')')'$$



用异或  
门实现



$$Y = A \oplus B$$



## 电路功能描述

穷举法

1

## 真值表

2

## 逻辑表达式

**例：**用与非门设计一个举重裁判表决电路。设举重比赛有3个裁判，一个主裁判和两个副裁判。杠铃完全举上的裁决由每一个裁判按一下自己面前的按钮来确定。只有当两个或两个以上裁判判明成功，并且其中有一个为主裁判时，表明成功的灯才亮。

1

设主裁判为变量A，副裁判分别为B和C；表示成功与否的灯为Y，根据逻辑要求列出真值表。

A	B	C	Y	A	B	C	Y
0	0	0	0	1	0	0	0
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	1	1

2

$$Y = m_5 + m_6 + m_7 = AB'C + ABC' + ABC$$



3  
↓  
卡诺图

化简  
↓

4  
最简与或  
表达式

5  
↓  
逻辑变换

6  
↓  
逻辑电  
路图

3  
↓

AB					
C		00	01	11	10
	0			1	
	1			1	1

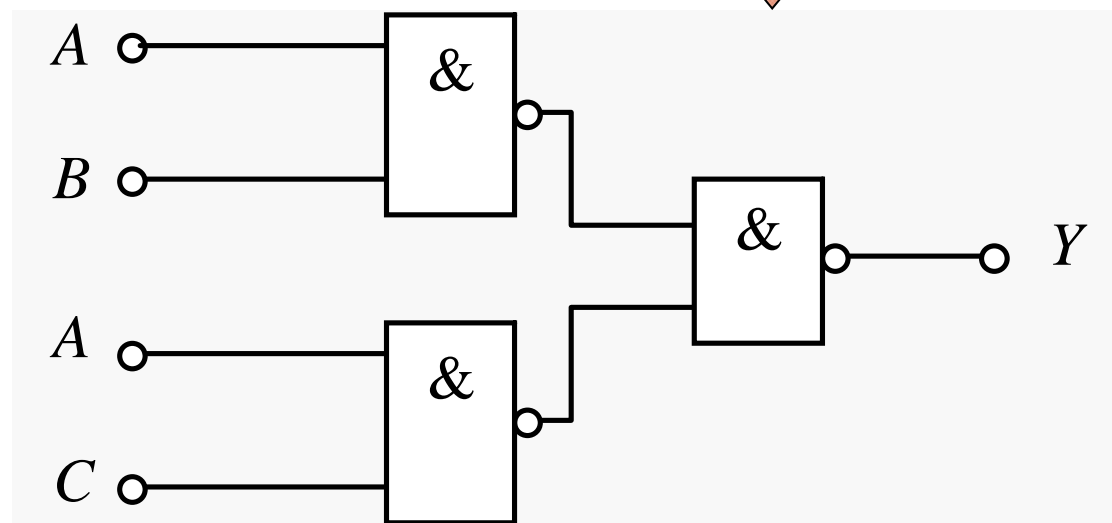
化简  
↓

4

$$Y = AB + AC$$

5

$$Y = ((AB)'(AC)')'$$



## 利用中规模组件设计组合电路(标准化设计)

中规模组件都是为了实现专门的逻辑功能而设计，但是通过适当的连接，可以实现一般的逻辑功能。

用中规模组件设计逻辑电路，可以减少连线、提高可靠性。

下面介绍用选择器和译码器设计组合逻辑电路的方法。

## 1. 用译码器实现组合逻辑函数

原理：变量译码器输出能产生输入变量的所有最小项。

高电平输出时：  $Y_i = m_i$

低电平输出时：  $Y_i' = m_i'$

而任何一个组合逻辑函数都可以变换为最小项之和的标准形式。因此，用译码器和门电路可实现任何单输出或多输出的组合逻辑函数。

当译码器输出低电平有效时，一般选用与非门；

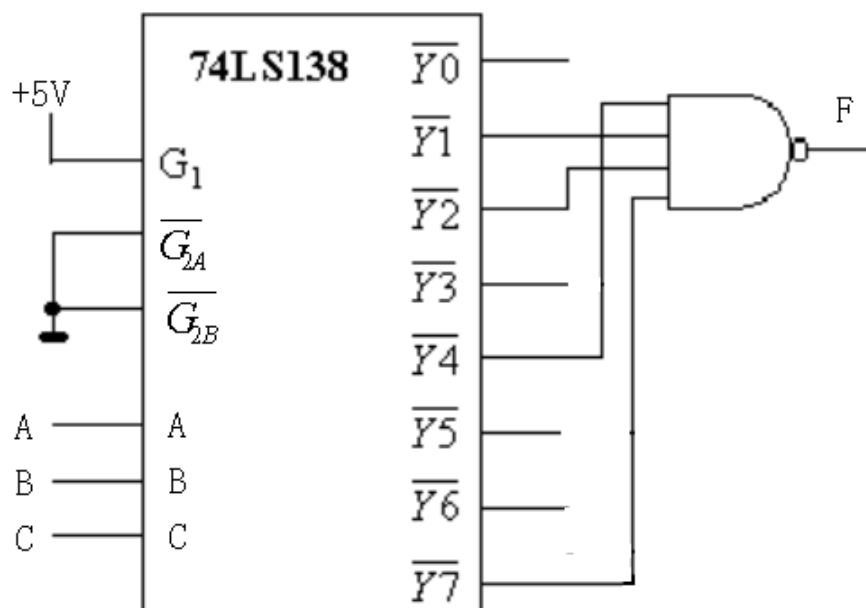
当译码器输出高电平有效时，一般选用或门；

## 例1 用译码器实现三个输入变量函数

$$F = \sum m(1,2,4,7)$$

解：译码器没有特指的情况下，指的都是变量译码器。低电平输出有效的译码器能产生输入变量的所有最小项的非。由于任何逻辑函数都可以按照最小项之和表示成标准积之和的形式，再二次求反，变成与非-与非式。因此可以想象，利用译码器得到最小项之非，而由外部的与非门来形成与非，即可实现逻辑函数。

由于本题有三个输入变量，总共有八个最小项。可以采用3线-8线译码器（如74LS138），得到逻辑电路图如下图所示。



$$F = m_1 + m_2 + m_4 + m_7$$

$$= m_1 \cdot m_2 \cdot m_4 \cdot m_7$$

## 例2 用译码器和门电路实现逻辑函数：

$$Y = A'B'C + ABC' + C$$

解：1) 选择译码器。

由于Y中有3个变量A、B、C，故应选3-8译码器，如74LS138。

因74LS138输出为低电平有效，故选用与非门。

2) 将Y变换为标准与或表达式。

$$Y = A'B'C + ABC' + ABC + AB'C + A'BC$$

$$= m_1 + m_3 + m_5 + m_6 + m_7$$

$$= m_1' \bullet m_3' \bullet m_5' \bullet m_6' \bullet m_7'$$

3) 令 $A_2=A$ 、 $A_1=B$ 、 $A_0=C$ ，可画出逻辑电路图。

例3 用译码器设计一个一位全加器。它能将两个二进制数及来自低位的进位进行相加，并产生和数与进位数。

解：1) 分析设计要求，列出真值表。

设被加数为 $A_i$ ，加数为 $B_i$ ，来自低位的进位为 $C_{i-1}$ 。

本位和为 $S_i$ ，向高位的进位为 $C_i$ 。

输 入			输 出	
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

## 2) 写出逻辑函数表达式

$$S_i = A_i' B_i' C_{i-1} + A_i' B_i C_{i-1}' + A_i B_i' C_{i-1}' + A_i B_i C_{i-1}$$

$$C_i = A_i' B_i C_{i-1} + A_i B_i' C_{i-1} + A_i B_i C_{i-1}' + A_i B_i C_{i-1}$$

## 3) 选择译码器

全加器有3个输入信号，有两个输出信号，因此可选74LS138和两个与非门来实现。

4) 将 $A_i$ 连 $A_2$ 、 $B_i$ 连 $A_1$ 、 $C_{i-1}$ 连 $A_0$ ，则 $S_i$ 、 $C_i$ 式变为：

$$S_i = (Y_1' \bullet Y_2' \bullet Y_4' \bullet Y_7')'$$

$$C_i = (Y_3' \bullet Y_5' \bullet Y_6' \bullet Y_7')'$$

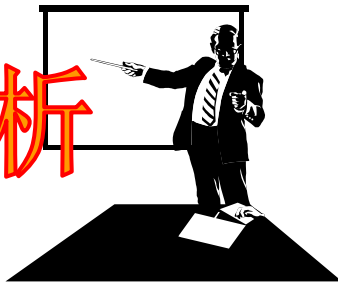
5) 有此可画出所设计的全加器电路图。

## 2、用数据选择器设计逻辑电路

四选一选择器功能表

输入			输出
$A_1$	$A_0$	$S'$	$W$
$\phi$	$\phi$	1	0
0	0	0	$D_0$
0	1	0	$D_1$
1	0	0	$D_2$
1	1	0	$D_3$

分析



$S' = 0$ 时:

$$W = D_0(A_1' A_0') + D_1(A_1' A_0) + D_2(A_0' A_1) + D_3(A_1 A_0)$$

类似三变量函数的表达式!



**例：**利用四选一选择器实现如下逻辑函数。

$$Y = R' A' G' + R' A G' + R A' G + A G$$

变换

$$Y = R'(G' A') + R'(G' A) + R(G A') + 1 \cdot (G A)$$

与四选一选择器输出的逻辑式比较

$$W = D_0(A_1' A_0') + D_1(A_1' A_0) + D_2(A_0' A_1) + D_3(A_1 A_0)$$

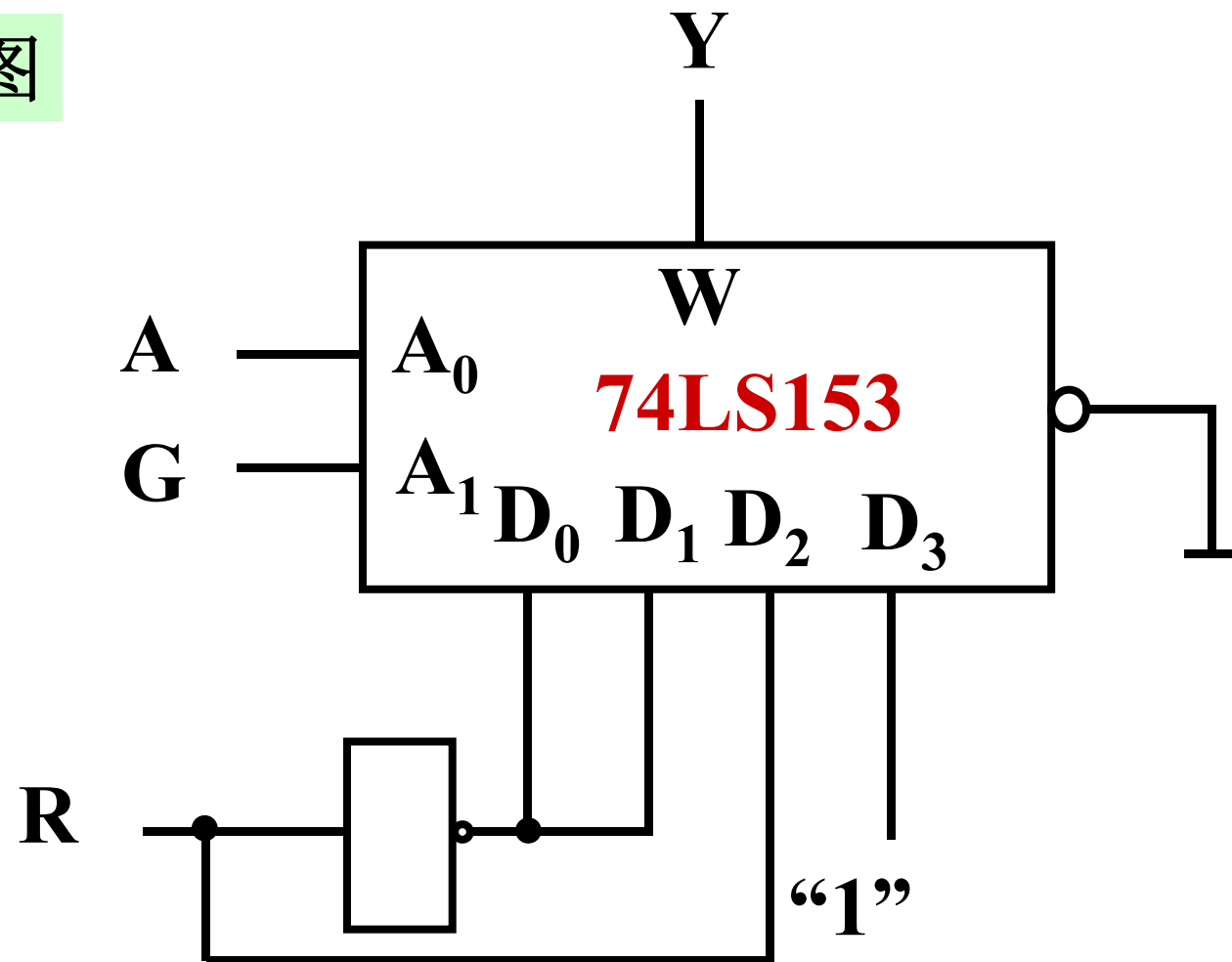
可以令：

$$\mathbf{G = A_1} \quad \mathbf{A = A_0}$$

$$\mathbf{D_2 = R} \quad \mathbf{D_3 = 1}$$

$$D_0 = D_1 = R'$$

# 接线图



$$Y = R' A' G' + R' A G' + R A' G + A G$$